TIBCO Spotfire® Server 6.5

Installation and Configuration Manual

Revision date: April 2014

Important Information

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

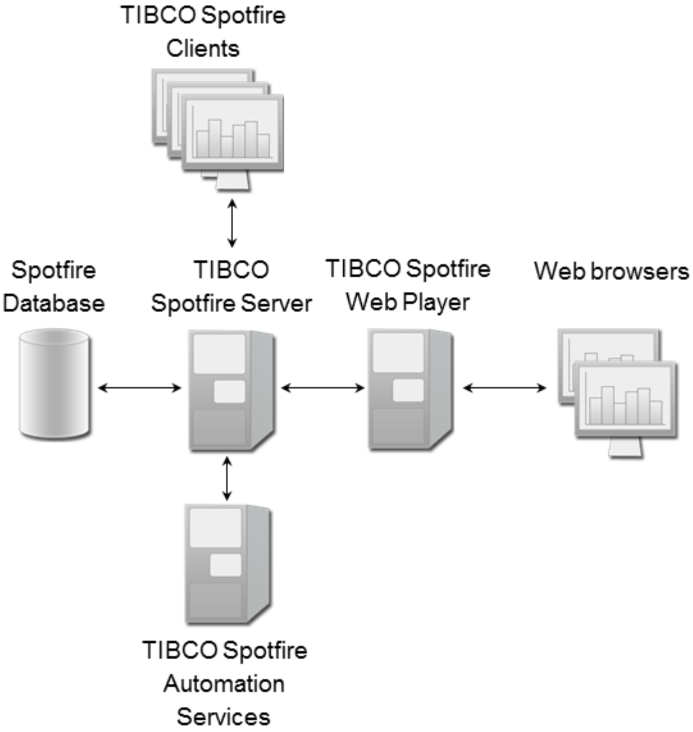
THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

TIBCO Software Inc. Confidential Information

Contents

1 Overview

The Spotfire® Server is used by TIBCO Spotfire®, TIBCO Spotfire® Web Player, and TIBCO Spotfire® Automation Services to identify users, assign privileges, and serve as a central storage for configuration, preferences, software updates, and analyses.



Spotfire®, Spotfire® Web Player, and Spotfire® Automation Services are all clients to the Spotfire® Server.

When starting Spotfire for the first time, an administrator needs to log in to a Spotfire Server and set up licenses for users. The users must activate their licenses to be able to access analyses in the Spotfire Server Library.

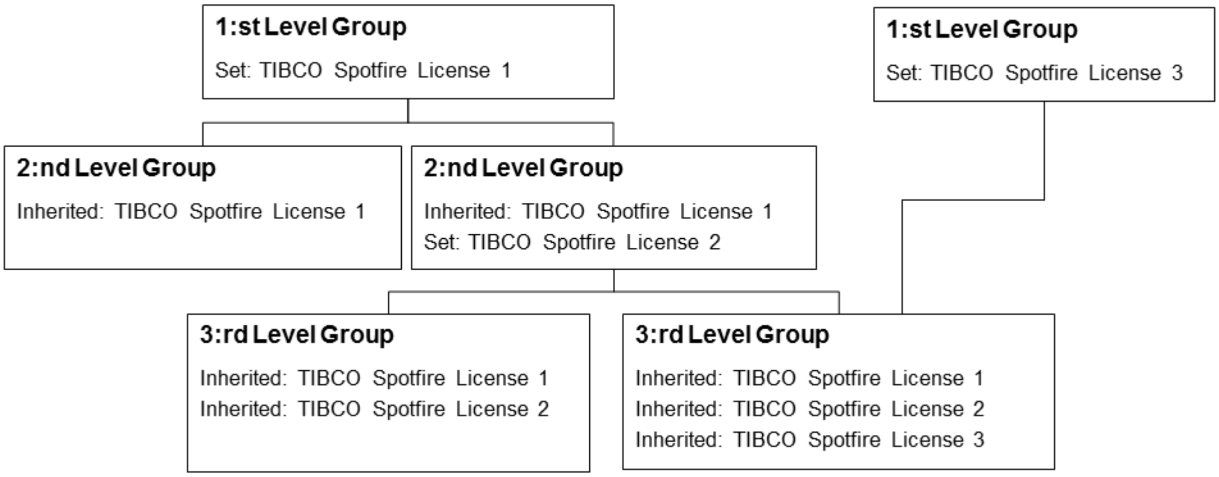
Note: For new or changed features, functionality changes, and information about issues, see the "TIBCO Spotfire Server - Release s" at http://docs.tibco.com.

1.1 Concepts

Groups

Spotfire administrators manage groups in the **Spotfire Administration Console** in Spotfire Server or in the **Spotfire Administration Manager** in Spotfire.

Groups can be structured into hierarchies where members of a child group are automatically members of all parent groups further up in the hierarchy. Properties, licenses, and preferences set for parent groups are inherited to child groups.


1:st Level Group
Set: TIBCO Spotfire License 1
1:st Level Group
Set: TIBCO Spotfire License 3
2:nd Level Group
Inherited: TIBCO Spotfire License 1
2:nd Level Group
Inherited: TIBCO Spotfire License 1
Set: TIBCO Spotfire License 2
3:rd Level Group
Inherited: TIBCO Spotfire License 1
Inherited: TIBCO Spotfire License 2
3:rd Level Group
Inherited: TIBCO Spotfire License 1
Inherited: TIBCO Spotfire License 2
Inherited: TIBCO Spotfire License 3

Spotfire Groups can be synchronized from an external source and can be created and managed locally in the Spotfire database. Synchronized groups cannot be managed from within the Spotfire system, but they can be placed in manually created groups and thereby managed.

Roles

In the Spotfire System, there are a number of **special groups** that are not possible to remove; these groups shall be seen as **roles**. Roles represent sets of tasks that the group members are allowed to perform. To assign a role to a user, add the user (or a group that the user belongs to) to one of the **special groups** listed below.

- Administrator
- Library Administrator
- Deployment Administrator
- Diagnostics Administrator
- Web Player Administrator
- Scheduled Updates Users
- Script Author
- Everyone
- Impersonator

Licenses

Spotfire licenses control which features are available to users. A group structure is created and licenses are set for the created groups. Licenses are always set on the group level. Groups can belong to groups, and licenses are inherited from parent groups.

When an end user logs in to Spotfire, only the features enabled on the groups the user belongs to appear in Spotfire.

Library

The **Spotfire library** provides convenient publishing of Spotfire files and analysis data. In the Spotfire library, users can publish and share their analysis material.

A Spotfire administrator or a library manager sets up the library using the **Administration Manager** in Spotfire. A folder structure is built and appropriate read and write permissions for different groups are assigned to the created folders.

Packages and Deployment

The features and functionality of Spotfire are contained in software packages. Many packages can be bundled together into a distribution. In the installation kit for the client you can find a distribution which contains the software for Spotfire.

Distributions and packages can be uploaded to a deployment area on the server; every user will be associated with a deployment area. When the user logs in, a check is made to see if there is updated content in the deployment area, and if so, the user is prompted to upgrade. The deployment area must have content, otherwise Spotfire will not work.

Different software can be uploaded to different deployment areas on Spotfire Server. This provides the possibility to decide which functionality a user will get. An administrator can create deployment areas, decide which software they will contain, and assign them to user groups.

1.2 Services

Spotfire Server handles requests and provides user data to Spotfire clients from the Spotfire database. The information Spotfire Server needs to have continuously is stored in the Spotfire database.

User Directory Services

Spotfire Server User Directory Services enables the server administrator to manage users, groups, licenses, and preferences stored in the Spotfire database. User Directory Services also performs user authentication and authorization. Spotfire Server can be integrated with an existing IT infrastructure.

Authentication:

- Checking credentials against:
  - Spotfire database
  - LDAP Directory
  - Windows NT Domain
  - Custom JAAS Module
- Single sign-on:
  - NTLM
  - Kerberos
  - X.509 Client Certificate

User Directory options:

- Spotfire database
- LDAP Directory
- Windows NT Domain

Deployment Services

Spotfire Server Deployment Services is the distribution mechanism for Spotfire client packages and hotfixes. It enables the creation and uploading of Packages to Spotfire Server.

The Deployment Services enables the upgrading of Spotfire clients from a central repository. When Spotfire is started, it checks for available updates and, if found, users are required to confirm download and install the latest version. The packages are stored in the Spotfire database.

Information Services

Spotfire Server **Information Services** accesses and transforms analysis data from data sources. Users can create Information Links to connect to external JDBC databases and thereby access and load data into Spotfire analysis files. Information Links and the elements they are created from are stored in the Spotfire database.

The supported data sources may be databases of various types, for example Oracle, SQL Server, DB2, MySQL, SAS, and Teradata.. Not explicitly supported databases may work if JDBC drivers exist and additional configuration is performed.

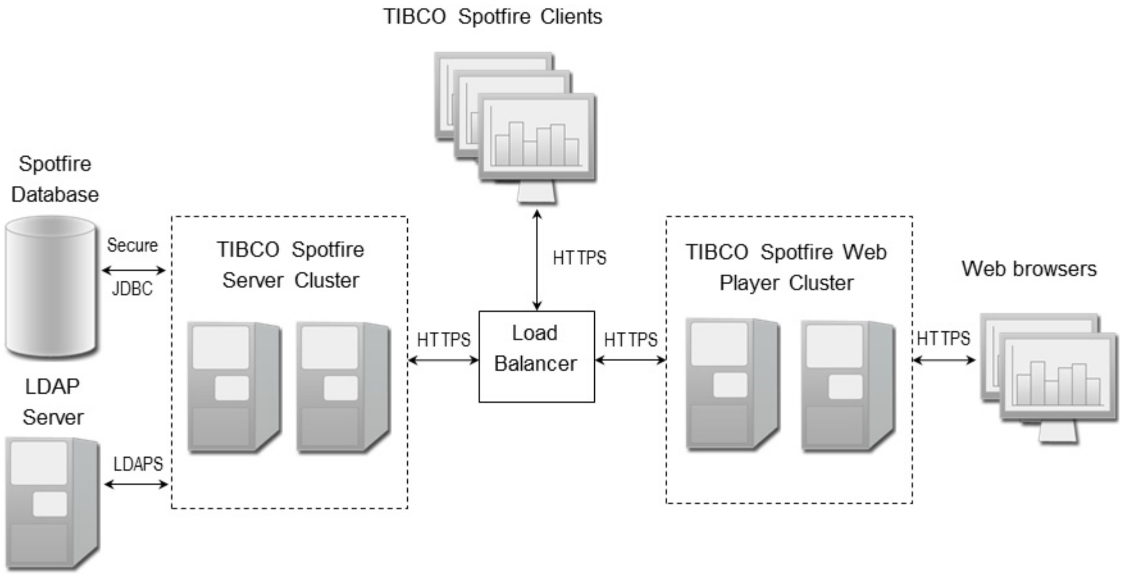
Library Services

Spotfire Server **Library Services** enables end users to share analyses in a central repository for analysis files. The library also stores other items needed to perform analyses. When a user selects **Save to Library**, the analysis file is stored in the Spotfire database and can be shared with other users. Access to files stored in the Library is controlled by user access permissions.

Spotfire users can save Spotfire analyses either to a local file system or to the Library, but analysis files must be saved to the Library for users to be able to open them in the Spotfire Web Player.

1.3 System

An advanced Spotfire system setup may include clustering as well as encryption of the communication using HTTPS, LDAPS, and Secure JDBC:



See “Authentication and User Directory” on page 55, “HTTPS” on page 95, and “Configuring LDAPS” on page 99 for details on security options.

Spotfire Server Clustering

The main purposes for clustering Spotfire Servers are fail-over support and load balancing. The load balancer must support *session affinity*, which means that within the cluster of servers, requests from the same client are always routed to the same server.

Each server stores its connection information locally and then reports its availability in the cluster by writing to the database. The first server that writes to the database marks itself as the *primus* server; this makes it responsible for tasks that should only be handled by one server per cluster, for instance synchronization of external LDAP users and groups. If this server goes down, another server will automatically take over the status as *primus*.

The database is the mechanism that enables clustering. In a clustered setup all servers use the same database. For security and performance reasons it is recommended not to install a Spotfire Server on the same machine as the database, not even in a non-clustered system. Spotfire Server connects to its database – Oracle or Microsoft SQL Server – using a JDBC driver.

Configuration

Spotfire Server configuration is performed by a graphical Configuration Tool or a Configuration Command Line Tool. The Configuration Tool guides the user through the configuration process and provides test features and feedback during configuration. The Configuration Command Line Tool enables scripting and provides advanced configuration capabilities.

In the tools, each configuration is stored as a specific object with a specific name, and is added to a list of configurations that includes timestamps and comments. This enables administrators to switch between configurations and also to test different configurations.

For information about the Configuration Tool, refer to section:

- "Configuration Tool" on page 34

An introduction to commands and their usage is given in the following sections:

- "Configuration Command Line Tool" on page 41
- "Creating a Simple Configuration Using the Configuration Command Line Tool" on page 43
- "Scripting a Configuration" on page 53
- "Authentication and User Directory" on page 55
- "Available Configuration Commands" on page 45 and the reference "Commands" on page 193

1.4 Spotfire Server Setup

Spotfire Server is set up in four general steps: preparation, installation, configuration, and administration. Actions indicated as recommended or optional may be performed later. Scripts, installers, tools, and applications are provided on the installation kit.

Preparation

- *Collect the required information* (page 15)
- *Prepare the Spotfire database* (page 17)

Installation

- *Run the Spotfire Server installer, interactively (using interface) or silently (using batch file)* (page 23)
- Recommended: *Install database drivers* (page 27)
- *Install the latest hotfix, if any* (page 27)

Configuration

- *Create a Spotfire Server configuration* (Simple, page 43)
- Recommended: *Further configure the Spotfire System* (page 55)
- Optional: *Set up load balancing* (See page 183, *Load Balancing Reference Implementation*)

Administration

- *Start Spotfire Server* (page 110)
- *Define Users and Groups* (page 112)
- *Deploy client packages to Spotfire Server* (page 113)
- *Install Spotfire for the Spotfire Administrator's usage* (page 114)
- *Assign Licenses and define Preferences* (page 114)
- *Install Spotfire clients on end users' machines* (page 114)
- Recommended: *Enable use of the data functions* (page 114)

  To use these data functions, TIBCO Spotfire® Statistics Services needs to be installed and configured as well.

  Spotfire 6.5 features pre-packaged predictive analytic methods in the form of data functions. These data functions provide Spotfire users immediately useful analytic functionality, as well as detailed and flexible templates to help users develop their own data functions more quickly and easily. Installing these data functions will make these features potentially available to your users.
- Optional: *Enable use of the Demo Database* (page 116)

1.5 Tuning

Temp Folder

Spotfire Server has a **temp folder** that is used to keep temporary files during uploads and downloads from Spotfire Server. It is recommended that this temp folder **<installation dir>\tomcat\temp\** be excluded from anti-virus checks.

Ports

Spotfire Server listens to the following ports; some may have to be modified:

Listener	Default Port
Spotfire Server (HTTP)	80
Spotfire Server (HTTPS)	Disabled (443)
Spotfire Server (JMX)	Disabled (1099)
Spotfire Server (AJP)	Disabled (8009)
Tomcat Shutdown	9005

Note: The Tomcat shutdown port is used to control the life cycle of the Spotfire Server application. This port is used locally and must not be exposed through the firewall.

1.6 Spotfire Server Upgrade

If you are running Spotfire Server 3.0 or later, you can perform an upgrade to Spotfire Server 6.5 using the Upgrade Tool distributed with Spotfire Server. The tool enables the 6.5 server to upgrade the database, and it copies the configurations from the previous server installation. See “Upgrading” on page 141.

1.7 System Requirements

Prior to installing Spotfire Server, verify that your system complies with the latest system requirements at **http://support.spotfire.com/sr.asp**.

Refer to the *Spotfire Server 6.5 Release s* for changes in this version.

2 Preparation

Upgrading?
If you are upgrading, first read “Upgrading” on page 141.

2.1 Collect the Required Information

Identify and **record** the system properties prior to installing Spotfire Server 6.5. Below is a checklist to help Administrators to identify the required information.

Note: When installing Spotfire Server, you will run scripts distributed with the installer. The scripts will create and prepare a database. See “Prepare the Database” on page 17.

Database server

Database server type: ____________________

There must be a database server up and running before installing Spotfire Server. The Spotfire Server installer will not install a database server. It is recommended that the database server is run on a dedicated, separate machine. Spotfire supports Oracle or Microsoft SQL Server.

Database server hostname: ____________________

Database server administrator username: ____________________

Database server administrator password: ____________________

SID (Oracle)/**Instance name** (MSSQL)**:**

Spotfire Database

When you set up the Spotfire database, you will create databases and database users.

Spotfire database name (MSSQL only): ____________________

The Spotfire database name only applies to MSSQL, and the default is **spotfire_server***.*

Spotfire database username:

If the database uses Windows Integrated Authentication, note this user here. If you use Integrated Authentication, Spotfire Server must run as this Windows Domain user.

Spotfire database password:

Spotfire Server

Spotfire Server Listen port:

The Spotfire Server installer will ask for a server listen port. The default is 80. If you are installing Spotfire Server on a machine that is already running other applications, make sure not to select a port that is already in use by these applications. The default port number is the same as in previous versions of Spotfire Server. If you want to continue to run a previous version of Spotfire Server on the same machine, select a different port number. Selecting a port number already in use may result in failure to start the servers. Refer to "Ports" on page 14 and "Basic Troubleshooting Steps" on page 300 for details.

Spotfire Server Login method:

Knowledge about your organization's IT infrastructure is required to set up any login method other than Spotfire Database.

- *Username and password login*

Valid options: Spotfire Database, LDAP, Custom JAAS, Windows NT Domain

- *Single sign-on login*

Valid options: NTLM, Kerberos, X.509 Client Certificate

Refer to "Authentication and User Directory" on page 55 for details.

Spotfire Server User Directory mode:

Knowledge about your organization's IT infrastructure is required to set up any user directory mode other than Spotfire Database. Valid options are: Spotfire Database, LDAP, and Windows NT Domain. Refer to "Authentication and User Directory" on page 55 for details.

Spotfire Server operating system:

Spotfire Server(s) Hostname(s):

Optional: **Hostname of load balancer:**

2.2 Prepare the Database

Before you run the Spotfire Server installer, you must set up the Spotfire database. This is performed by modifying scripts and running them.

You can use either an Oracle database server or a Microsoft SQL Server to hold your Spotfire Server database. Depending on your choice, continue to either:

- “Oracle” on page 17
- “Microsoft SQL Server” on page 19

2.2.1 Oracle

2.2.1.1 Prerequisites

The following settings must be configured on the Oracle Server in order for the scripts to work, and for Spotfire Server to be able to communicate with the databases:

- The Oracle Server must be configured to use username and password authentication to run the scripts. However, it is possible to set up Spotfire Server to authenticate with an Oracle database instance using Kerberos. See the section “Using Kerberos to Log In to the Spotfire Database” on page 100.
- National Language Support (NLS) must be set to match the language in which you will store data (affects search).

If the database server NLS cannot be set to match the language in which you will store data, Oracle provides other methods of setting NLS to a specific database or user, such as per session. Talk to your database administrator or refer to the Oracle database documentation for more information.

2.2.1.2 Copy the Scripts to the Database Server

The scripts run commands that needs to communicate with the Spotfire database. Therefore it is recommended to copy the content below to the database server. Use any machine that can run Oracle tools and can communicate with the database server.

1. On the installation kit, locate the directory **scripts/oracle_install**
2. Copy the entire directory to a temporary place on the local disk of your intended database server.

Comment: The command line database tools (sqlplus etc.) must be in the system path of the database server.

2.2.1.3 Modify the create_databases Script

In the copied folder, locate **create_databases.bat** (Windows) or **create_databases.sh** (Solaris/Red Hat Enterprise Linux/SUSE Linux Enterprise).

1 Open the file in a text editor and locate the following section:

```
rem Set these variable to reflect the local environment:
set ROOTFOLDER=<ROOTFOLDER>
set CONNECTIDENTIFIER=<SID>
set ADMINNAME=system
set ADMINPASSWORD=<ADMINPASSWORD>
set SERVERDB_USER=<SERVERDB_USER>
set SERVERDB_PASSWORD=<SERVERDB_PASSWORD>
set SERVER_DATA_TABLESPACE=SPOTFIRE_DATA
set SERVER_TEMP_TABLESPACE=SPOTFIRE_TEMP

rem Demo data parameters
set INSTALL_DEMODATA=no
set DEMODB_USER=spotfire_demodata
set DEMODB_PASSWORD=spotfire_demodata
```

2 Specify the following variables.

- ROOTFOLDER. The location where the tablespaces will be created. It must be a directory that is writable for the Oracle server, usually **<oracle install dir>/oradata/<SID>**. **Note:** Do not add a slash or backslash after the **<SID>**.
- CONNECTIDENTIFIER: The Oracle TNS name/SID of the database.
- ADMINNAME: The name of a user with database administrator privileges, default is **system account**.
- ADMINPASSWORD: The password of the above user.
- SERVERDB_USER: The user that will be created and later used to access the Spotfire database.
- SERVERDB_PASSWORD: The password of the above user.
- SERVER_DATA_TABLESPACE: The name of the tablespaces that will be created. The default value will work for most systems.
- SERVER_TEMP_TABLESPACE: The name of the temporary tablespaces that will be created. The default value will work for most systems.

  Note: Conflicting tablespaces can occur if you are creating the Spotfire tablespaces on a database server that is already hosting an Analytics Server or a previous version of Spotfire Server. Make sure that you do not select any names for the 6.5 tablespaces and users that conflicts with the already hosted tablespaces and users.

- INSTALL_DEMODATA: If you need demodata, alter this variable from default “no”. The demo database contains example data for learning about Spotfire. Do not change the default username. If you install the demo database, you must perform additional steps to make the data available to the users: See “Enable Demo Database Usage” on page 116.

3 Save the file and exit the text editor.

2.2.1.4 Run the create_databases.bat/.sh Script

Once the scripts have been set up, run the **create_databases.bat** or **create_databases.sh** script. Running these scripts will execute the other SQL scripts in the folder.

1 Open a command prompt window.

2 Navigate to the directory where you placed the scripts.

3 Type **create_databases.bat** or **create_databases.sh** and press **Enter**.

Response: The scripts now set up the Spotfire database required by Spotfire Server, and optionally the Spotfire Demo Database. This may take a while.

Response: The **log.txt** file is created in the same directory as the **create_databases** file. Also, if you selected to have demodata, a number of log files from the creation of the Spotfire demo data will be created. Please examine these files to verify that no errors occurred. Please retain the logs for future reference.

Note: There is sensitive information in the scripts. It is recommended to remove the scripts after they have run.

Proceed to “Install Spotfire Server” on page 23 when finished.

2.2.2 Microsoft SQL Server

There are different scripts provided for Microsoft SQL Server. Which to select depends on if you intend to set up Windows Integrated authentication or username and password authentication between Spotfire Server and the Spotfire database:

- For username and password authentication on Microsoft SQL Server, use **create_databases.bat**.
- For Windows Integrated authentication (IA) on Microsoft SQL Server, use **create_databases_ia.bat**.

2.2.2.1 Prerequisites

The following settings must be configured on the Microsoft SQL Server in order for the scripts to work, and for Spotfire Server to be able to communicate with the databases:

- TCP/IP communication must be enabled.
- A TCP/IP listener port must be configured. (The default is 1433.)
- A case insensitive collation must be used (at least for the Spotfire database).
- Collation must match the language in which you will store data (affects search).

2.2.2.2 Copy the Scripts to the Database Server

The scripts run commands that needs to communicate with the Spotfire database. Therefore it is recommended to copy the content below to the database server. Use any machine that can run Microsoft SQL tools and can communicate with the database server.

1 In the installation kit, locate the scripts directory: **scripts/mssql_install**

2 Copy the entire directory to a temporary place on the local disk of your intended database server.

Comment: The command-line database tools (sqlcmd, and so forth) must be in the system path of the database server.

2.2.2.3 Modify the create_databases.bat Script

In the copied folder locate **create_databases.bat** or **create_databases_ia.bat**.

1 Open the file in a text editor and locate the following section:

In create_databases.bat:

```
rem Set these variable to reflect the local environment:
set CONNECTIDENTIFIER=<SERVER>\<MSSQL_INSTANCENAME>
set ADMINNAME=sa
set ADMINPASSWORD=<ADMINPASSWORD>
set SERVERDB_NAME=spotfire_server
set SERVERDB_USER=<SERVERDB_USER>
set SERVERDB_PASSWORD=<SERVERDB_PASSWORD>
```

In create_databases_ia.bat:

```
rem Set these variable to reflect the local environment:
set CONNECTIDENTIFIER=<SERVER>\<MSSQL_INSTANCENAME>
set WINDOWS_LOGIN_ACCOUNT=<WINDOWS_LOGIN_ACCOUNT_DOMAIN>\
<WINDOWS_LOGIN_ACCOUNT_NAME>
set SERVERDB_NAME=spotfire_server
set SERVERDB_USER=<SERVERDB_USER>
```

Both files also contain a demo data section:

```
rem Demo data parameters
set INSTALL_DEMODATA=no
set DEMODB_NAME=spotfire_demodata
set DEMODB_USER=spotfire_demodata
set DEMODB_PASSWORD=spotfire_demodata
```
(only in create_databases.bat)

Note: The demo data includes sample data and analysis files, from a variety of fields, that showcase some of Spotfire's capabilities. Users can expand their Spotfire skills by working with these examples.

2 Specify the following variables that apply to the authentication method you are using:

- CONNECTIDENTIFIER: Replace **<SERVER>** with the name of the server running the SQL Server instance, and replace **<MSSQL_INSTANCENAME>** with the name of the SQL Server instance.
- ADMINNAME: The name of a user with admin privileges on the database, usually "**sa**". (This is not used if you are using Windows Integrated Authentication.)

- ADMINPASSWORD: The password of the above user. (This is not used if you are using Windows Integrated Authentication.)
- WINDOWS_LOGIN_ACCOUNT: The Windows Login Account that is used to authenticate with the database server. (This is only used when using Windows Integrated Authentication.)
- SERVERDB_NAME: The name of the database that will be created.
- SERVERDB_USER: The user that you wish to create and use for this database.
- SERVERDB_PASSWORD: The password for the above user. (This is not used if you are using Windows Integrated Authentication.)
- INSTALL_DEMODATA: If you want to install demo data, change this variable from the default "no" to "yes". If you want to create the demo database, do not change the default database name and username. Make sure that the database server does not have a database or a user with these names already. If you install the demo database, you must perform additional steps to make the data available to the users. See "Enable Demo Database Usage" on page 116.

Note: When using Windows Integrated Login, the **create_server_user_ia.sql** script creates a database user associated with the WINDOWS_LOGIN_ACCOUNT name. By default, it is assumed that a Windows login with this name already exists. If it does not exist, and you wish to create such a login, open the script in a text editor and uncomment the section that reads:

```
/*
use master
GO
CREATE LOGIN [$(WINDOWS_LOGIN_ACCOUNT)] FROM WINDOWS WITH
DEFAULT_DATABASE=[$(SERVERDB_NAME)], DEFAULT_LANGUAGE=[us_english]
GO
ALTER LOGIN  [$(WINDOWS_LOGIN_ACCOUNT)] ENABLE
GO
DENY VIEW ANY DATABASE
TO  [$(WINDOWS_LOGIN_ACCOUNT)]
*/
```

3 Save the file and exit the text editor.

Case Sensitive Collation

If your database server by default is set to use a **case sensitive** collation, you must specify that the Spotfire database shall be case insensitive. Edit the SQL script **create_server_db.sql**:

1 Open this file in a text editor.

2 Locate the commented out line:

```
--create database $(SERVERDB_NAME) collate Latin1_General_CI_AS;
```

3 Remove the leading "--". Set the collation to the collation of your preference, and make sure it is case insensitive (CI), for example **Latin1_General_CI_AS**. Refer to the Microsoft SQL Server documentation for more information about available collations.

4 Comment out the line below it by inserting leading “--”:

```
create database $(SERVERDB_NAME)
```

5 Save the file and exit the editor.

2.2.2.4 Run the create_databases Script

Once the scripts have been set up, run the **create_databases.bat** or **create_databases_ia.bat** script. Running this script will execute all of the other sql scripts in the folder.

1 Open a command prompt window.

2 Navigate to the directory where you placed the scripts.

3 Type **create_databases.bat** or **create_databases_ia.bat** and press **Enter**.

Response: The scripts now set up the database table needed to run Spotfire Server. that this may take some time.

Response: A number of log files will be created in the same directory as the **create_databases** file. Please examine these files to verify that no errors occurred. Please retain the logs for future reference.

Note: There is sensitive information in the scripts. It is recommended to remove the scripts after they have run.

Proceed to “Install Spotfire Server” on page 23 when finished.

3 Installation

This chapter describes the installation of Spotfire Server including post-installation steps. Running the installer, both in interactive and silent mode, is described.

> **Upgrading?**
> If you are upgrading, first read “Upgrading” on page 141.

3.1 The First Installation Process

1. **Run the Spotfire Server installer, interactively or silently** (3.2)
2. *Recommended*: **Install Database Drivers** (3.3)
3. **Install the latest hotfix, if any** (3.4)

3.2 Install Spotfire Server

Note: Spotfire Server is no longer supported on 32-bit systems.

In the installation kit, locate the installer for your system:

- Windows: **setup-win64.exe**
- RPM Linux: **tss-6.5.0.x86_64.rpm**
- Tarball Linux: **tss-6.5.0.x86_64.tar.gz**
- Solaris: **install.bin**

Three major components will be installed: A Java Environment (JDK), a Tomcat Application Server and a Spotfire Server web application. The JAVA_HOME of the Apache Tomcat is set to the path of the installed JDK.

Note: Consider installing the Java Cryptography Extension (JCE) unlimited strength jurisdiction policy files. It is the user's responsibility to verify that these files are allowed under local regulations.

Windows Specifics

The native Windows installer can be run interactively or silently.

RPM Linux Specifics

The RPM based Linux installer is launched from the command line. The post-install script must then also be launched from the command line. This script accepts arguments on the command line, or will ask for them if they are missing.

Note: The Spotfire Administrator must have **root** access to install the RPM based Spotfire Server on Linux.

Tarball Linux Specifics

Tarball Linux can be used if the Spotfire Administrator does not have root access.

The Tarball Linux installer is unpacked and launched from the command line. The post-install script must then also be launched from the command line. This script accepts arguments on the command line, or will ask for them if they are missing.

Solaris Specifics

Spotfire Server is installed on Solaris using the InstallAnywhere technology, which is also used by previous Spotfire Server versions.

JCIFS for NTLM Authentication

Third party components are needed to configure the system for NTLM, these can be automatically downloaded as a step in the installation, or manually downloaded at a later point.

To manually download the files, go to **http://public.tibco.com/pub/tibco_oss/jcifs/jcifs_1.3.17.zip** and copy **jcifs_1.3.17.zip (**which contains the file **jcifs.jar)** to the **<installation directory>/tomcat/webapps/spotfire/WEB-INF/lib/** folder.

3.2.1 Interactive Installation

Windows

1 Run the selected installer.

Note: If you use Microsoft SQL Server with Windows Integrated Authentication, install Spotfire Server as the Domain User that you set up with the script **create_databases_ia.bat**. Also make sure that Spotfire Server always runs as this Domain User, see “Windows, Service Exists, Integrated Authentication for SQL Server” on page 111. Confirm with the logs that Spotfire Server starts.

2 The *Welcome* dialog is displayed.
Click **Next**.

3 The *TIBCO License* dialog is displayed. Read the license agreement and select the appropriate radio button.
Click **Next**.

4 The *Third Party Components* dialog is displayed.
These components are only needed to configure the system for NTLM. See “JCIFS for NTLM Authentication” on page 24.
Click **Next**.

If you select to download third party components, the *Third Party License* dialog is displayed. Read the license agreement and select the appropriate radio button.
Click **Next**.

5 The *Destination Folder* dialog is displayed. Specify the Spotfire Server location.
Click **Next**.

6 The *Windows Service* dialog is displayed. Select the option you want.
Click **Next**.

7 The *Spotfire Server Port* dialog is displayed. Specify the Spotfire Server port, for details See “Ports” on page 14.
Click **Next**.

8 The *Ready to Install the Program* dialog is displayed.
Click **Install**.

Response: The installation starts.

9 The *Completed* dialog provides three options:

- **Launch the configuration tool**: Select this option to start configuring Spotfire Server using the Configuration Tool, See “Configuration” on page 29.
- **Launch the upgrade tool**: Select this option if you are performing an upgrade from a previous version of Spotfire Server and want the upgrade tool to start automatically when the installer finishes.
- **Exit the Installation Wizard.**

Select the **Show the Windows Installer log** option to see the log when the installer finishes.

Click **Finish**.

Solaris

1 Run the installer.

2 The *Introduction* dialog is displayed.
Click **Next**.

3 The *License* dialog is displayed. Select a radio button.
Click **Next**.

4 The *Third Party Components* dialog is displayed.
These components are only needed to configure the system for NTLM. See “JCIFS for NTLM Authentication” on page 24.
Click **Next**.

If you selected to download third party components in the previous dialog, the *Third Party License* dialog is displayed. Read the license agreement and select the appropriate radio button.
Click **Next**.

5 The *Installation Folder* dialog is displayed. Specify the install location.
Click **Next**.

6 Spotfire Server *Port* dialog is displayed. Specify the port, See “Ports” on page 14.
Click **Next**.

7 The *Installation Summary* dialog is displayed.
Click **Install**.

8 The *Upgrade* dialog is displayed. If you are performing an upgrade from a previous version of Spotfire Server, select to start the upgrade tool automatically when the installer finishes.
Click **Next**.

9 The *Install Complete* dialog is displayed.
Click **Done**.

3.2.2 Command Line Installation

Windows

To launch a silent installation, run the installer as follows:

```
setup-win64.exe /s /v"/qn /l*vx c:\setup.log DOWNLOAD_THIRD_PARTY=Yes
INSTALLDIR=c:\tibco\tss\6.5.0 SPOTFIRE_WINDOWS_SERVICE=Create SERVER_PORT=80"
```

Arguments:

- DOWNLOAD_THIRD_PARTY: Can be one of {**Yes**, **No**}.
  See “JCIFS for NTLM Authentication” on page 24.
- SPOTFIRE_WINDOWS_SERVICE: Can be one of {**Create**, **DoNotCreate**}.
  See step 6 on page 24.
- Specify **/qn** for quiet installation with no user interface.
- Specify **/qb** for quiet installation with basic user interface.

RPM Linux

To launch a silent installation, run **rpm -ivh tss-6.5.0.x86_<64>.rpm**

Use the post-install configuration arguments:

```
/user/local/bin/tibco/tss/6.5.0/configure [-d] [-s <server port>]
```

where **-d** disables the download of third party components.

Tarball Linux

Download the tar file and unpack it in the desired directory using the command:
tar xzf tss-6.5.0.x86_64.tar.gz.

Use the post-install configuration arguments:

```
./configure [-d] [-s <server port>]
```

in the directory where the tar file was unpacked, where **-d** disables the download of third party components.

If needed, configure the server to start on boot by running the command:
./configure-boot

Note: To run this specific command, the Spotfire Administrator must have **root** access.

Solaris

To create a silent installation, first record a response file. To record, type: **install.bin -r <path>/silent.properties** and specify a path to the file, where the path should be an absolute path.

To run the installer silently: **install.bin -i silent -f <path>/silent.properties**. If no path is specified to the properties file, it is assumed it is located in the same folder as the **install.bin**.

3.3 Install Database Drivers

Spotfire Server ships with DataDirect database drivers. While these drivers work well for test environments as described above, it is strongly recommended that drivers from your database server vendor are used in a production environment. Vendor drivers (JDBC) are available to download from Microsoft's and Oracle's homepages. Place them in the **<installation dir>/tomcat/lib** directory. When you have installed the database drivers, Spotfire Server needs to be restarted according to the instructions in the section "Start and Stop Spotfire Server" on page 110.

Note: The database connection URL, used by the server to connect to the database, may differ for different database drivers. See "Database Connection URL Examples" on page 181 for the database connection URL and for database driver examples.

Data Sources in the Information Designer

The Information Designer tool, available in Spotfire, allows users to create analyses based on data retrieved from external JDBC sources. These external data sources are accessed using database drivers. Any database drivers you install in the folder mentioned above are available for use in the Information Designer.

To connect to an external data source, you also need to enable a data source template that matches the database and a specific database driver. Use the Configuration Tool or the command "add-ds-template" on page 193.

3.4 Hotfix Installation

Before you continue, please check if there is a relevant hotfix for this version of Spotfire Server. If there are, install the latest, every hotfix is cumulative, which means that you only need to install the latest.

1. Check the URL http://support.spotfire.com/patches_spotfireserver.asp and download the latest hotfix. Installation instructions for each hotfix are included in the package.
2. Always make sure you have installed the latest hotfix before troubleshooting or reporting any problems.

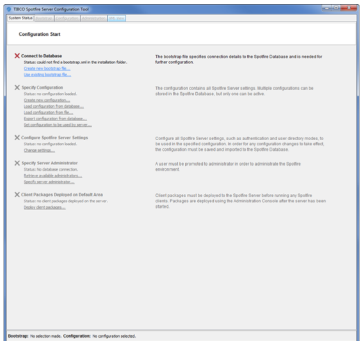
When you have installed a possible hotfix, the next step is to configure the Spotfire system. See "Configuration" on page 29.

4 Configuration

Spotfire Server configuration

Spotfire Server is configured using:

1 **Configuration Tool**, started using **uiconfig.bat**/**uiconfig.sh** or


TIBCO Spotfire Server Configuration Tool
System Status
Bootstrap
Configuration
Administration
XML View
Create new bootstrap file...
Use existing bootstrap file...
Create new configuration...
Load configuration from database...
Load configuration from file...
Export configuration from database...
Set configuration to be used by server...
Change settings...
Retrieve available administrators...
Specify server administrator...
Deploy client packages...

Configuration Start

✗ Connect to Database
Status: could not find a bootstrap.xml in the installation folder.

The bootstrap file specifies connection details to the Spotfire Database and is needed for further configuration.

✗ Specify Configuration
Status: no configuration loaded.

The configuration contains all Spotfire Server settings. Multiple configurations can be stored in the Spotfire Database, but only one can be active.

✗ Configure Spotfire Server Settings
Status: no configuration loaded.

Configure all Spotfire Server settings, such as authentication and user directory modes, to be used in the specified configuration. In order for any configuration changes to take effect, the configuration must be saved and imported to the Spotfire Database.

✗ Specify Server Administrator
Status: No database connection.

A user must be promoted to administrator in order to administrate the Spotfire environment.

✗ Client Packages Deployed on Default Area
Status: no client packages deployed on the server.

Client packages must be deployed to the Spotfire Server before running any Spotfire clients. Packages are deployed using the Administration Console after the server has been started.

Bootstrap: No selection made. Configuration: No configuration selected.

2 **Configuration Command Line Tool**, executed using **config.bat**/**config.sh**. This tool enables configuration scripting and provides advanced configuration capabilities.

The Configuration Tool is presented in Section 4.2 on page 34 and an introduction to the Configuration Command Line Tool is presented in Section 4.3 on page 41.

When the Spotfire database has been set up, and the server is installed, a minimum amount of configuration is required to get a system running. (See “Creating a First Configuration” on page 31.)

Available commands are grouped by function in Section 4.3.3 on page 45 and detailed command documentation is listed alphabetically by command in the reference “Commands” on page 193.

4.1 High Level Configuration Steps

Configuring Spotfire Server is a three-step procedure. Also see the **TIBCO Spotfire Server Quick Start Guide** for basic steps.

1 **Create a bootstrap file**

The bootstrap file contains database connection information and is required to connect to the Spotfire database.

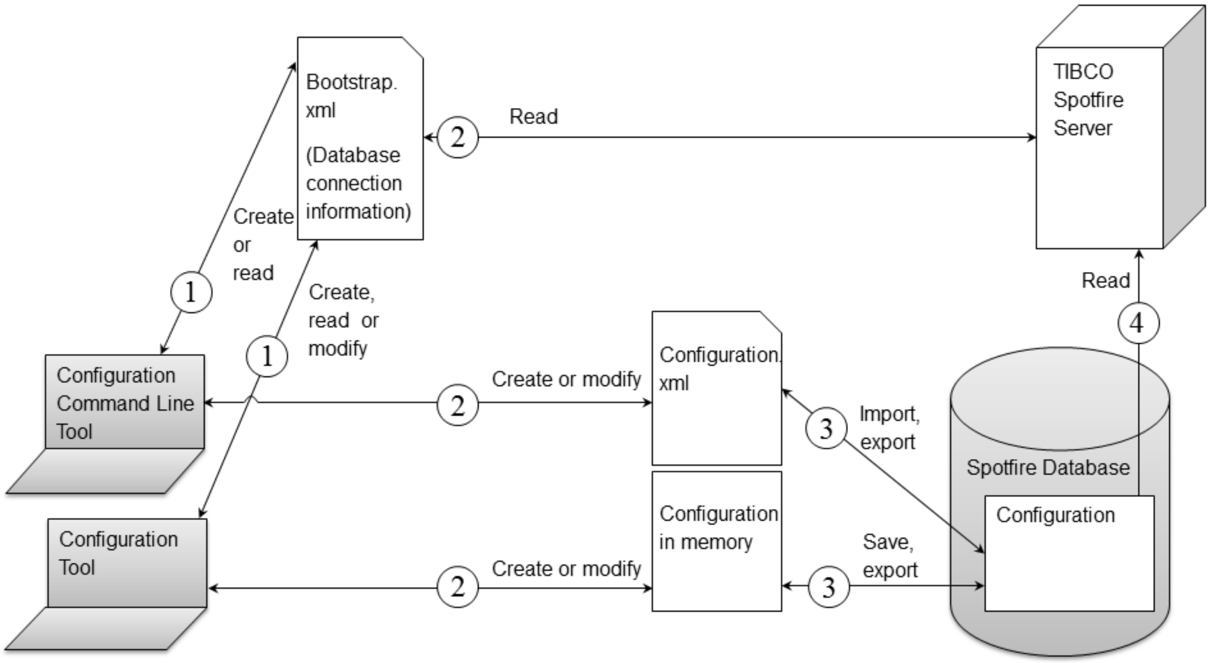
2 **Create a configuration**

A completed configuration contains settings for the authentication components, the User Directory, the Spotfire Library, etc. During the configuration procedure, the Configuration Tool keeps the configuration in memory and the Configuration Command Line Tool creates and modifies the configuration as a **configuration.xml** file.

3 **Save/Import the configuration**

The configuration has to be **saved** from the Configuration Tool, or **imported** from the Configuration Command Line Tool, to the Spotfire database.

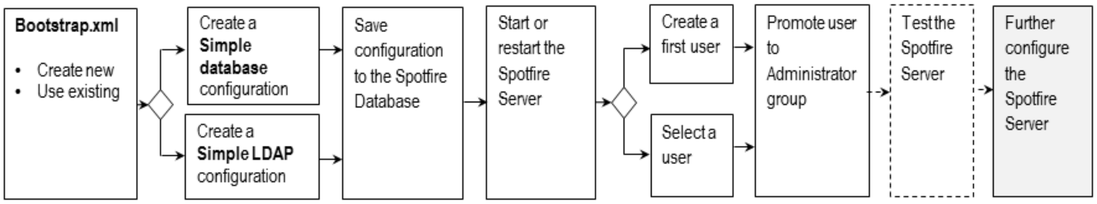
4 When Spotfire Server starts, it reads the configuration from the database.



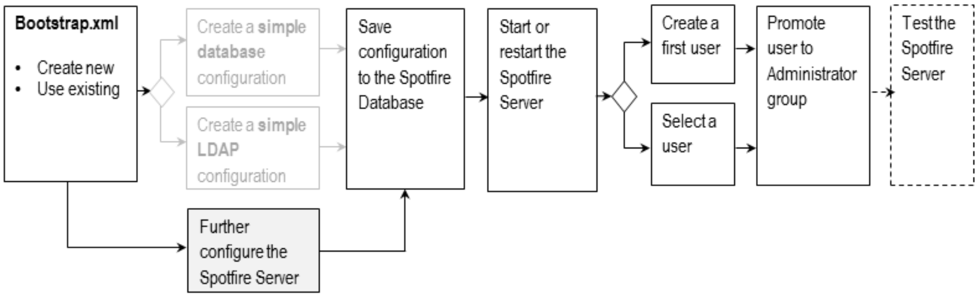
4.1.1 Creating a First Configuration

Spotfire Server provides a broad set of configuration options. The ones you will use depend on the installation environment, requirements, etc.

One alternative is to apply a configuration workflow where a simple database configuration or a simple LDAP configuration is set up, then tested, and then further configuration is performed.


Bootstrap.xml
• Create new
• Use existing
Create a Simple database configuration
Create a Simple LDAP configuration
Save configuration to the Spotfire Database
Start or restart the Spotfire Server
Create a first user
Select a user
Promote user to Administrator group
Test the Spotfire Server
Further configure the Spotfire Server

Another alternative is to apply a configuration workflow where all configuration is performed before testing the setup.


Bootstrap.xml
• Create new
• Use existing
Create a simple database configuration
Create a simple LDAP configuration
Further configure the Spotfire Server
Save configuration to the Spotfire Database
Start or restart the Spotfire Server
Create a first user
Select a user
Promote user to Administrator group
Test the Spotfire Server

Further configuration options are listed below:

Authentication
User name and Password (4.6)

- Spotfire database
- LDAP Directory
- Windows NT
- Custom JAAS

Single Sign On (4.7)

- NTLM
- Kerberos
- X.509 Certificates

Delegated Authentication (4.9)

Impersonation (4.10)

Post Authentication Filter (4.5.3)

- Blocking Mode
- Auto Creating Mode
- Chained Custom Filter

User Directory Modes

- Spotfire database (4.13.1)
- LDAP (4.13.2)
- Windows NT (4.13.3)

Login Behavior (4.11)

Additional Security

- HTTPS (4.14)
- LDAPS (4.15)
- LDAP SASL (4.6.3)
- Kerberos for Spotfire database (4.16)

Exporting and Importing Configuration Files (4.17)

Configuring a Specific Directory for Library Import and Export (4.18)

Load Balancing (page 185)

For information how to create a simple configuration in different ways, see sections:

- "Creating a Simple Configuration Using the Configuration Tool" on page 40. Also see the *TIBCO Spotfire Server Quick Start Guide*.
- "Creating a Simple Configuration Using the Configuration Command Line Tool" on page 43
- "Scripting a Configuration" on page 53

4.1.2 Configuration Procedure After the First Configuration

It is possible to update configurations using the Configuration Tool or the Configuration Command Line Tool.

It is also possible to configure Spotfire Server by editing a local copy of the **configuration.xml** file and then importing it, thereby setting it as the active configuration. Editing the xml-file can also be used to configure complex features in XML rather than by commands.

To edit the configuration.xml file

1 Export the **configuration.xml** file, see "Exporting and Importing Configuration Files" on page 107.

2 Open the **configuration.xml** file in a text editor.

3 Locate the element that contains the configuration property you want to change.

4 Change the configuration property.

5 Save the **configuration.xml** file.

6 Import the **configuration.xml** file, see "Exporting and Importing Configuration Files" on page 107.

7 Restart Spotfire Server.

▶ To update a server configuration in the Configuration Tool:

Note: The configuration Tool is described in "Configuration Tool" on page 34.

1 Open the Configuration Tool.

Connect to the Spotfire Database.
If you have already run the tool and created a bootstrap file placed in the **<installation dir>/tomcat/webapps/spotfire/WEB-INF** directory, the Configuration Tool will open this and prompt for its tool password.

Enter the tool password.

2 The active configuration is loaded into the Configuration Tool.

3 Make your changes to the configuration on the **Configuration** tab.

4 Import the configuration into Spotfire Server by clicking **Save**.

5 Start or restart Spotfire Server.

▶ To update a server configuration in the Command Line Configuration Tool:

Note: The Configuration Command Line Tool is described in "Configuration Command Line Tool" on page 41.

1 Open a command prompt. (In Windows, click **Start,** type **cmd** in the **Search** box, and then press Enter.)

2 Run the **export-config** command to export the configuration from the Spotfire database to a configuration file.

Example, where "configuration.xml" is optional and the **-f** (**--force**) is not applied:

```
> config  export-config  configuration.xml
```

3 Update the configuration in the configuration file using selected commands.

Example, where "**--configuration=configuration.xml**" is optional:

```
> config  config-auth  --configuration=configuration.xml  --auth-method=BASIC
--jaas-database
```

4 Choose restart policy and run the **set-restart-policy** command.

Example, where "**--configuration=configuration.xml**" is optional:

```
> config  set-restart-policy  --configuration=configuration.xml  --policy=
AUTOMATIC_ON_IDLE
```

5 Run the **import-config** command to import the updated configuration file into the Spotfire database.

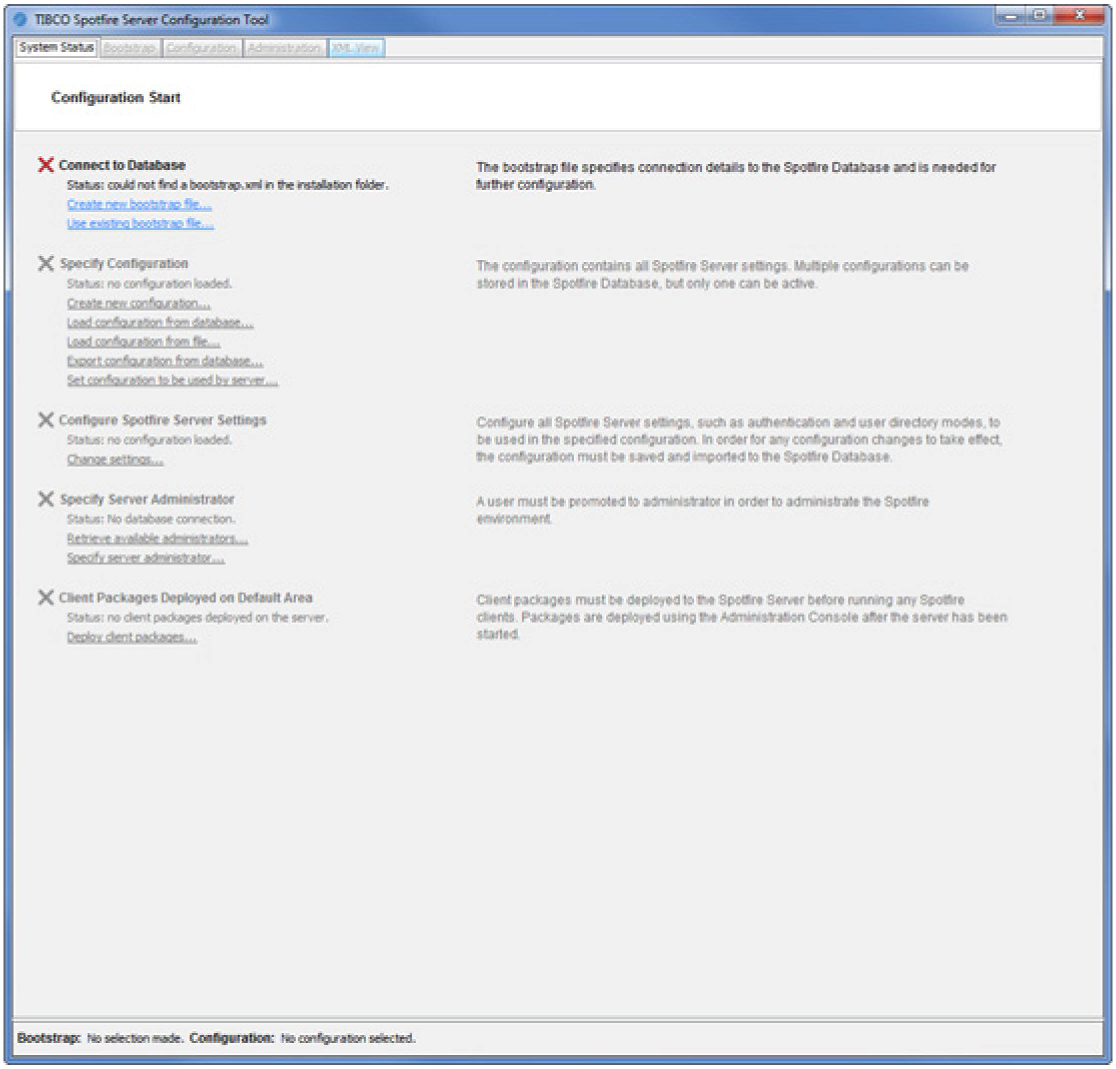
Example, where "configuration.xml" is optional:

```
> config  import-config  --comment="Switched to BASIC authentication using the Spotfire
Database authentication source"  configuration.xml
```

6 Optionally restart the server(s) if the restart policy is set to **MANUAL**.

7 Optionally remove the **configuration.xml** files or restrict the access permissions to it. **Note**: Do not remove the **bootstrap.xml** file. See "bootstrap.xml" on page 176.

4.2 Configuration Tool



Configuration Start

✗ Connect to Database
Status: could not find a bootstrap.xml in the installation folder.
Create new bootstrap file...
Use existing bootstrap file...

The bootstrap file specifies connection details to the Spotfire Database and is needed for further configuration.

✗ Specify Configuration
Status: no configuration loaded.
Create new configuration...
Load configuration from database...
Load configuration from file...
Export configuration from database...
Set configuration to be used by server...

The configuration contains all Spotfire Server settings. Multiple configurations can be stored in the Spotfire Database, but only one can be active.

✗ Configure Spotfire Server Settings
Status: no configuration loaded.
Change settings...

Configure all Spotfire Server settings, such as authentication and user directory modes, to be used in the specified configuration. In order for any configuration changes to take effect, the configuration must be saved and imported to the Spotfire Database.

✗ Specify Server Administrator
Status: No database connection.
Retrieve available administrators...
Specify server administrator...

A user must be promoted to administrator in order to administrate the Spotfire environment.

✗ Client Packages Deployed on Default Area
Status: no client packages deployed on the server.
Deploy client packages...

Client packages must be deployed to the Spotfire Server before running any Spotfire clients. Packages are deployed using the Administration Console after the server has been started.

Bootstrap: No selection made. **Configuration:** No configuration selected.

The Configuration Tool can be launched:

- From the start menu.
- By selecting **Launch the Configuration Tool** (last step in the installation wizard).
- By running the **uiconfig.bat** file (run **uiconfig.sh** on Linux). These files are located in the **<installation dir>\tomcat\bin** folder.

Always run the Configuration Tool as an administrator. When launching the Configuration Tool, the **System Status** tab is displayed.

Note: If there already is a **bootstrap.xml** file in the **<tomcat\webapps\spotfire\WEB-INF>** folder, a *Specify Tool Password* dialog is displayed. Enter the Configuration Tool password and click **OK** to unlock the bootstrap file (see “Bootstrap tab” on page 37 for information on the Configuration Tool Password). Click **Cancel** to close the dialog and display the **System Status** tab.

Configuration Tool on a Local Machine

If the Spotfire Administrator does not have access to Spotfire Server machine or if the server is incapable of displaying graphics, it is possible to run the Configuration Tool from a local machine. To do this, copy the JAR file **spotfireconfigtool.jar** to the local machine. It is located in the **<installation dir>/tomcat/webapps/spotfire/tools/** folder on the server. If Spotfire Server is up and running, it can also be downloaded by clicking on the link on the **Tools** page. Double-click the file or run the command **java –jar spotfireconfigtool.jar** to unpack it to the directory
../spotfireconfigtool and run the **uiconfig.bat** (or **uiconfig.sh**) file from there.

Note: When downloading the file spotfireconfigui.jar, the bootstrap.xml file, that has the necessary information to be able to connect to the database, is not included. You must either create a new bootstrap.xml file with the database connection information or copy an existing bootstrap file from the server. This is located in

<installation dir>/tomcat/webapps/spotfire/WEB-INF/bootstrap.xml

Note: To be able to run the Configuration Tool locally, a Java 7 runtime must be installed and present in the path on the local machine.

Configuration Tool Navigation

Navigate in the tool by clicking the links in the *System Status* tab and/or the tabs at the top. The Configuration Tool contains five tabs where selections can be made:

- **System Status** tab
- **Bootstrap** tab
- **Configuration** tab
- **Administration** tab
- **XML View** tab

System Status tab

This tab shows the status of the configuration and presents five **tasks**. Complete the tasks from top to bottom. The tasks are preceded by a green check mark or a red cross.

- Green check mark: a valid alternative is selected for the task.
- Red cross: a valid alternative is not selected for the task.
- The tasks can be modified regardless of symbol. A greyed out task means that a preceding task is not completed.

The tasks are followed by shortcut links to the place where the setting is altered in the Configuration Tool. The **Client Packages Deployed on Default Area** task can also be performed in the Spotfire Administration Console.

The five tasks in the **System Status** tab are presented below:

1 Connect to Database.

This task deals with bootstrap files. The **bootstrap.xml** file contains database connection information. Click the appropriate link to create a new bootstrap file or use an existing file.

2 Specify Configuration.

The configuration contains all Spotfire Server settings. Multiple configurations can be stored in the Spotfire database, but only one can be active.

Note: If there is an active configuration in the Spotfire database, the Configuration Tool will try to load it.

Click the appropriate link under the Specify Configuration task to specify the configuration you want to work on in the following tasks 3-5:

- To create a new configuration, click **Create new configuration**.
- To modify an existing configuration located in the Spotfire database, click **Load configuration from database**.
- To select an existing configuration and set it active, click **Set configuration to be used by server**.

For experienced users who want to edit xml:

- To modify an existing configuration from file, click **Load configuration from file**.
- To export a configuration from the Spotfire database, click **Export configuration from database**.

3 Configure Spotfire Server Settings

Click the link to access the **Configuration** tab.

4 Specify Server Administrator.

A user must be promoted to administrator in order to administrate the Spotfire environment. Click the **Specify Server Administrator** link to access the **Administration** tab and perform the task.

5 Client Packages Deployed on Default Area.

Click the link to deploy client packages to the default deployment area. **Note:** This function should only be used to deploy the Spotfire Client packages that are required for all installations. To deploy any additional packages, use the Administration Console.

Note: In the Spotfire Administration Console, Client packages can be deployed to any of the existing deployment areas in the Spotfire database (not only the default deployment area). If a deployment exists in an area other than the default, this task will be marked with a red cross even though a valid deployment exists.

You can create, delete and rename deployment areas and also change the default deployment area in the Spotfire Administration Console; see "Deploy Spotfire" in the Spotfire Administration Console Help for more information.

Bootstrap tab

The **Bootstrap** tab has two alternative presentations: **Create new Bootstrap file** and **View Bootstrap File**. To create a new bootstrap file, enter information in the fields and click **Create Bootstrap**.

Path The bootstrap file must be located in the default path for Spotfire Server to be able to use it.

Database server Select from drop-down menu.

JDBC driver Select from drop-down menu.

Driver Class The name of the JDBC driver class. This field is pre-populated from selections made and cannot be edited.

Database username The name of the database account used by potfire Server to connect to the Spotfire database. Enter correct database login details.

Database password The password of the database account. Enter correct database login details. This field is not relevant when using Windows Integrated Authentication.

Hostname Enter the database hostname.

Port Enter the Spotfire database port.

SID (for Oracle) Enter Server ID.

Database name (for MS SQL) Enter the database name.

Database URL The JDBC connection URL. This field is pre-populated from selections made and can be edited.

Configuration Tool Password Enter a Configuration Tool password of your own choice. This will be used to protect the server configuration from unauthorized access. **Note:** The Configuration Tool password will be required when running the Configuration Tool now and later.

Server Name Enter a unique name for Spotfire Server of your own choice.

Encryption Password (optional) Enter an encryption password of your own choice. This will be used for encrypting other passwords stored in the database. that the passwords are encrypted with a static key if no encryption password is specified here. Also note that in a clustered environment, the same encryption password must be specified for all servers in the cluster.

Note: Some fields are pre-populated, and most of these can also be altered manually.

Configuration tab

Configure all Spotfire Server settings to be used in the specified configuration. In order for any configuration changes to take effect, the configuration must be saved to the Spotfire database, this is done by clicking **Save configuration**.

If the Spotfire database already has been configured and contains an active configuration, this will be loaded into the Configuration Tool. If this is the case, navigate to the **System Status** tab and click the **Create new configuration** link. This will create a new default configuration and show it in the Configuration Tool.

Available panels:

Authentication See “Authentication and User Directory” on page 55.

User Directory See “User Directory Modes” on page 80.

Domain See “External Directories and Domains” on page 77.

Authentication Filter Here you can configure a standard servlet filter that can be used for various forms of authentication. All non-authorized requests that go to Spotfire Server will go through this filter. To use the filter for authentication, "Authentication Filter" must be specified as the source type in the External Authentication panel.

If the user is authenticated by the filter, it should wrap the HttpServletRequest and override the method getUserPrincipal() to return the user name of the authenticated user, before calling doFilter on the filter chain with the original response and the wrapped request object.

In the panel, you should specify the Filter class; the fully qualified name of the class implementing the javax.servlet.Filter interface. You can also specify optional initialization parameters that will be provided to the filter.

For more information on servlet filters, see <

http://docs.oracle.com/javaee/6/api/javax/servlet/Filter.html.

Post Authentication Filter See “Authentication and User Directory” on page 55.

External Authentication See “External Authentication Method” on page 73.

Impersonation See “Impersonation” on page 75.

Monitoring (JMX) See “Monitoring” on page 117.

Restart Policy

This panel is used to set the way the server reacts to configuration changes. Each server periodically checks for configuration changes and handles any such changes according to the Restart Policy.

MANUAL: changes do not have any effect until the server(s) are manually restarted

AUTOMATIC_FORCE: The server(s) are immediately automatically restarted.

AUTOMATIC_ON_IDLE: The server(s) are automatically restarted when considered idle.

Note: The AUTOMATIC_FORCE option may result in currently running user operations being aborted.

Login Dialog

See "Configuring Login Behavior" on page 76.

Data Source Templates

See "Data Source Templates" on page 158.

Attachment Manager

See "Attachment Manager" on page 108.

User Action Log

See "Action Logs and System Monitoring" on page 121.

Database Properties

This panel is used to modify the common configuration for the connection to Spotfire Server database. This configuration (which affects all servers) will be merged with the configuration in the **bootstrap.xml** file on each server.

Join Database

This panel is used to configure the default join database. If a join is made between data residing on two (or more) different datasources an attempt is made to copy the data from one of the datasources to the other to perform an SQL join. If the involved datasources do not allow writing then the join is performed using the default join data source.

Custom JAAS

This panel is used to create a custom JAAS configuration. If Custom JAAS is selected as authentication the created configurations will be available from the Authentication panel.
See "Authentication towards a Custom JAAS Module" on page 60.

Library Directory	This panel is used to configure the library import/ export directory. All library import and export operations will be done from/to this directory which may be a local directory or reside on a shared disk.

See referenced chapters for details. Navigate between the settings in the tree structure in the upper-left corner in the Configuration Tool.

Administration tab

Spotfire Server needs an Administrator.

- If configuring an LDAP User Directory or a Windows NT User Directory, search for users, select a user and click **Promote**.
- If not using LDAP, enter user name and password in the **Create new user** fields and click **Create**. Select the user and click **Promote**.

XML View tab

This tab displays the content of the tool's currently loaded configuration in XML format. The tab is read-only.

4.2.1 Creating a Simple Configuration Using the Configuration Tool

1 Start the Configuration Tool.

2 On the **System Status** tab, click **Create new Bootstrap**.

Response: The **Bootstrap** tab is displayed.

3 Enter information in the fields. See “Bootstrap tab” on page 37.

Click **Create bootstrap**

Response: The Configuration Tool checks that database drivers are installed and that the database is running. It also checks that the database accepts the given credentials. A dialog is displayed showing successful or unsuccessful creation of the bootstrap file.
On successful creation of the bootstrap file, the *System Status* tab of the Configuration Tool is displayed.

4 Click the **Configuration** tab.

5 Review the configuration. Verify that **BASIC Database** is selected in the Authentication drop-down menu and that **Database** is selected in the User Directory drop-down menu.

Click **Save configuration**.

6 The Save Configuration wizard is displayed and **Database** is pre-selected.

Click **Next**.

Response: The "To import the configuration you must provide a comment" dialog is displayed.

7 Enter a comment, for instance "Initial configuration". Click **Finish**.

8 Go to the **Administration** tab in the Configuration Tool.

9 In the **Create new user** section, type the user name and password that you would like to use for your Spotfire Administrator account

Click **Create**.

Response: The **Created new user** window opens.

10 Click **OK**.

11 Select the new user name from the Users column and click **Promote** to add that user to the Administrators group.

4.3 Configuration Command Line Tool

To configure Spotfire Server using the Configuration Command Line Tool run **config.bat** on Windows (available in the folder **<installation dir>\tomcat\bin**, which for a default Windows installation is **C:\tibco\tss\6.5.0\tomcat\bin)**. On Linux the tool is named **config.sh**. To execute a command, type **config.sh <command>**.

The Configuration Command Line Tool can be used in two ways. Either execute commands one by one in a console; the section "Creating a Simple Configuration Using the Configuration Command Line Tool" on page 43 uses this method. Or create a script containing several commands that are executed in one go, using the **run** command; the section "Scripting a Configuration" on page 53 details that procedure. Both methods can be used to configure Spotfire Server after a simple configuration.

The section below first describes how to launch a command prompt, how to move in the directories, and how to run the Configuration Command Line Tool. Next it explains how to execute the **help** command for the **version** command, and then execute the simple **version** command.

Note: Paths and comments using spaces must be enclosed in *straight* quotes ("). More advanced editors may change straight quotes to smart quotes, resulting in errors when running the commands.

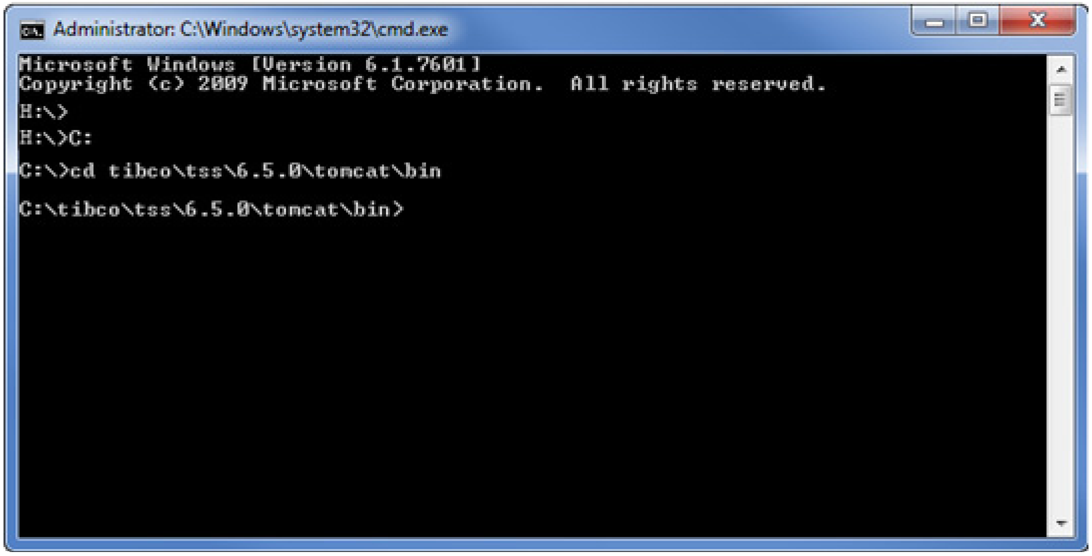
Note: The user running the Configuration Command Line Tool needs write access to the **tomcat\logs** folder.

▸ Executing a Configuration Command:

1 To execute a Configuration Command, select **Start > Command Prompt**, or select **Start**, type **cmd** in the *Search programs and files* input field and select **cmd.exe**. Start the command prompt as an administrator when running the script.

2 Change folder of the Configuration Command Line Tool **config.bat/config.sh** by first writing **cd**, then enter the path to the directory of the Configuration Command Line Tool (or copy the path, right-click at the cursor in the window and select **Paste**) and press **Enter**.

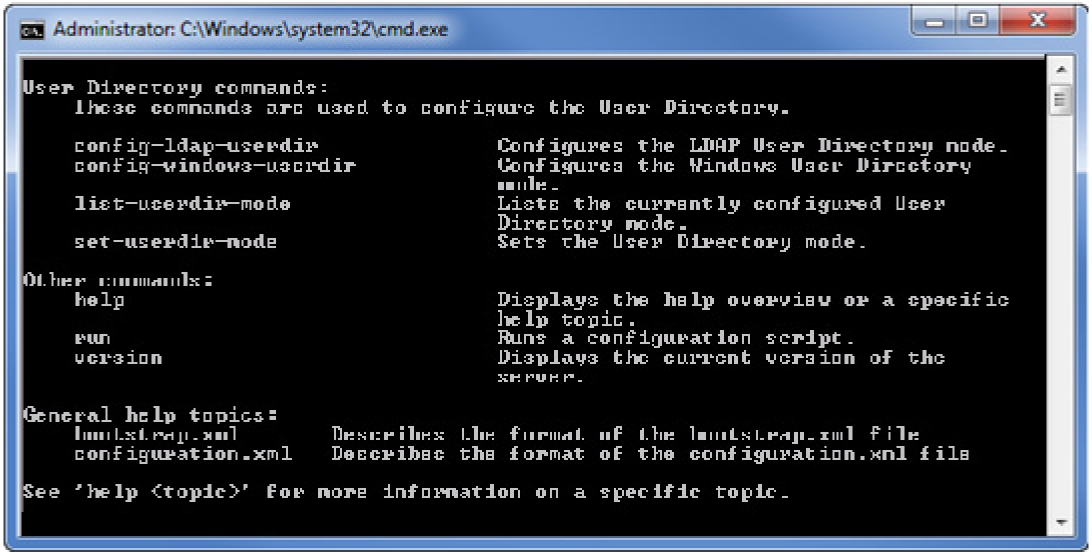
Response: This is where you execute commands in the Command Line Configuration Tool:.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.
H:\>
H:\>C:

C:\>cd tibco\tss\6.5.0\tomcat\bin

C:\tibco\tss\6.5.0\tomcat\bin>
```

3 Enter **config.bat** (Windows) or **config.sh** (Linux) and press **Enter**:

Response: The Spotfire Configuration Command Line Tool is run. Since it is run without command argument, it writes the list of available commands to the console and terminates:



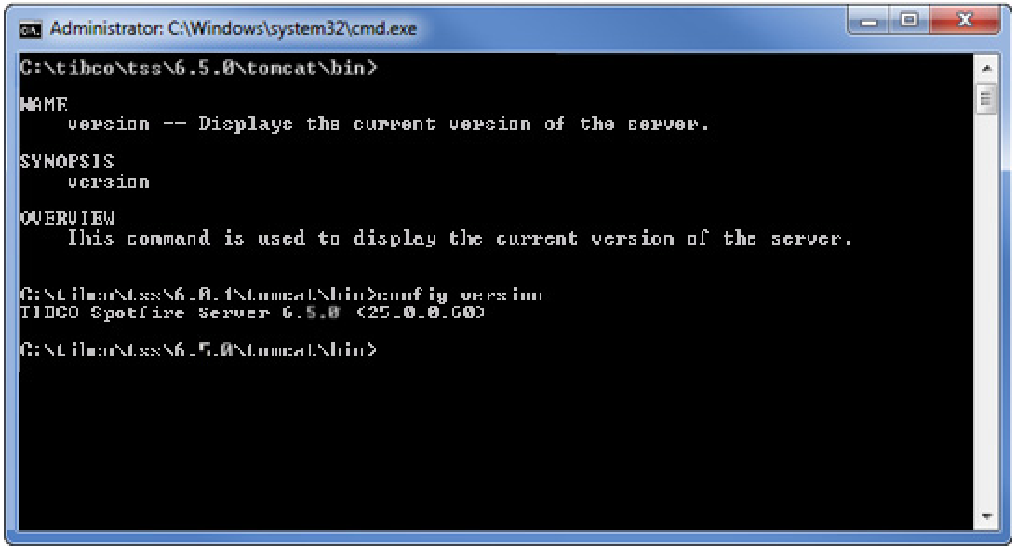
```
User Directory commands:
    These commands are used to configure the User Directory.

    config-ldap-userdir              Configures the LDAP User Directory node.
    config-windows-userdir           Configures the Windows User Directory
                                     mode.
    list-userdir-mode                Lists the currently configured User
                                     Directory mode.
    set-userdir-mode                 Sets the User Directory mode.

Other commands:
    help                             Displays the help overview or a specific
                                     help topic.
    run                              Runs a configuration script.
    version                          Displays the current version of the
                                     server.

General help topics:
    bootstrap.xml       Describes the format of the bootstrap.xml file
    configuration.xml   Describes the format of the configuration.xml file

See 'help <topic>' for more information on a specific topic.
```

Executing Commands:

To execute commands, type **config** followed by a command:

- To view the help topic on the **version** command, type **config help version** and press **Enter**. The tool is run executing the **help** command with the version parameter, and then terminates.

- To run the **version** command, type **config version** and press **Enter**. The tool is run executing the version command, writing the version information to the console:



```
C:\tibco\tss\6.5.0\tomcat\bin>

NAME
    version -- Displays the current version of the server.

SYNOPSIS
    version

OVERVIEW
    This command is used to display the current version of the server.


C:\tibco\tss\6.0.1\tomcat\bin>config version
TIBCO Spotfire Server 6.5.0 (25.0.0.60)

C:\tibco\tss\6.5.0\tomcat\bin>
```

Many commands require parameters, like the **create-default-config** command. Some are complex, like the **bootstrap** command. Both will be used in the next step, where a simple Spotfire Server configuration is created. But first section 4.3.3 lists the available commands.

4.3.1 Creating a Simple Configuration Using the Configuration Command Line Tool

Integrating Spotfire Server with an existing LDAP environment?

If you plan to integrate Spotfire Server with an existing LDAP environment, proceed to “Authentication and User Directory” on page 55. That section describes how to create a configuration with LDAP authentication and User Directory.

If you follow the instructions in this section and later the instructions in the LDAP section, the Spotfire database will be populated with both Spotfire database and LDAP accounts, where only the LDAP accounts will be possible to use. The instructions in the LDAP section do not build upon the instructions in this section.

A simple configuration with Spotfire database authentication and User Directory is the most basic setup of Spotfire Server. To create the simple configuration, use the Configuration Tool or the Configuration Command Line Tool.

To achieve this, run five commands in the Configuration Command Line Tool or run a prepared script, refer to “Scripting a Configuration” on page 53 to use a prepared script. The script provides variable definitions that can be edited and the variables are then used by the commands. Of course the commands can be run manually one after another, arguments to the commands can be seen in the script, also see section 4.3.2.

To execute a Command, open a command prompt and change working directory to **<installation dir>/tomcat/bin**. In a default Windows installation: **C:\tibco\tss\6.5.0\tomcat\bin**. Type **config help <command>** (on Linux: **config.sh help <command>**) to access the help for a particular command or refer to the “Commands” on page 193.

4.3.2 Manually Creating a Simple Configuration

To configure Spotfire Server and get it up and running the following is required (after the Spotfire Database has been set up, described in chapter 2, and Spotfire Server Installer has been run, described in chapter 3).

1 Create the connection configuration needed by the server to connect to the database with the **bootstrap** command.

If you have already run the **bootstrap** command, there is no need to run it again unless you want to use different arguments.

- **--driver-class**: The fully qualified class name of the JDBC driver
- **--database-url**: The JDBC connection URL
- **--username**: The name of the database account used by Spotfire Server to connect to the Spotfire database
- **--password**: The password of the database account
- **--tool-password**: Choose a Configuration Command Line Tool password that will be used to protect the server configuration from unauthorized access and/or modification

Replace the **<DRIVER CLASS>**, **<DATABASE URL>**, **<DATABASE USERNAME>**, **<DATABASE PASSWORD>** and **<CONFIG TOOL PASSWORD>** with the appropriate values.

```
> config bootstrap --driver-class="<DRIVER CLASS>" --database-url="<DATABASE URL>"
--username="<DATABASE USERNAME>" --password="<DATABASE PASSWORD>"
--tool-password="<CONFIG TOOL PASSWORD>"
```

Response: A **bootstrap.xml** file is created in the **<installation directory>\tomcat\webapps\spotfire\WEB\INF** folder. See “bootstrap.xml” on page 176.

Example:

```
> config bootstrap --driver-class="tibcosoftwareinc.jdbc.oracle.OracleDriver"
--database-url="jdbc:tibcosoftwareinc:oracle://MyDBServer:1521;SID=XE" --username=
"dbuser" --password="dbpwd" --tool-password="configtoolpwd"
```

2 Create a default configuration with the **create-default-config** command:

```
> config create-default-config
```

Response: A **configuration.xml** is created.

3 Import the configuration to the database to set it active with the **import-config** command.

Replace the **<CONFIG TOOL PASSWORD>** and **<DESCRIPTION>** with the appropriate values.

```
> config import-config --tool-password="<CONFIG TOOL PASSWORD>" --comment=
"<DESCRIPTION>"
```

Example: Import the configuration to the database

```
> config import-config --tool-password="configtoolpwd" --comment="First config"
```

4 Create a first user with the **create-user** command. This step will create an account that can be used to login to Spotfire Server.

Replace the **<CONFIG TOOL PASSWORD>**, **<SPOTFIRE ADMIN USERNAME> and <SPOTFIRE ADMIN PASSWORD>** with the appropriate values.

```
> config create-user --tool-password="<CONFIG TOOL PASSWORD>" --username=
"<SPOTFIRE ADMIN USERNAME>" --password="<SPOTFIRE ADMIN PASSWORD>"
```

Example: Creating a new user account

```
> config create-user --tool-password="configtoolpwd" --username="SpotfireAdmin"
--password="s3cr3t"
```

5 Add the first user to the Administrator group with the **promote-admin** command:

Replace the **<CONFIG TOOL PASSWORD>** and **<SPOTFIRE ADMIN USERNAME>** with the appropriate values.

```
> config promote-admin --tool-password="<CONFIG TOOL PASSWORD>" --username=
"<SPOTFIRE ADMIN USERNAME>"
```

Example: Promote the new account to administrator

```
> config promote-admin --tool-password="configtoolpwd" --username="SpotfireAdmin"
```

When Spotfire Server is running, the first Administrator can create other users and add them to the Administrator group using the Administration Console.

4.3.3 Available Configuration Commands

Commands are grouped below into functional areas for easy reviewing. The command parameters to use depend on the setup and the system environment. Review applicable commands and their parameters in the alphabetically ordered reference “Commands” on page 193 or by using the **help** command in the Configuration Command Line Tool.

Most configuration commands work towards the **configuration.xml** file. The file can be created using the **export-config** command and manually edited. The configuration is uploaded to the Spotfire database by the **import-config** command, using database connection information in the **bootstrap.xml** file created by the **bootstrap** command.

Some commands work directly towards the Spotfire database, like the Administration commands listed below.

4.3.3.1 Administration Commands

These commands are used to perform basic administration tasks. All administration commands connect directly to the database.

add-member	Adds a user or group as a member of a specified group
create-user	Creates a new user account
delete-disabled-users	Deletes disabled users

delete-disconnected-groups	Deletes disconnected groups
delete-user	Deletes a user account
demote-admin	Revokes full administrator privileges from a user
enable-user	Enables or disables a user in the Spotfire Database
export-groups	Exports groups from the User Directory
export-library-content	Exports content from the library
export-users	Exports users from the User Directory
import-groups	Imports groups to the User Directory
import-library-content	Imports content into the library
import-users	Imports users to the User Directory
list-admins	Lists the server administrators
list-deployment-areas	Lists the deployment areas
list-groups	Lists all groups
list-online-servers	Lists all online servers
list-users	Lists all users
manage-deployment-areas	Manages the deployment areas
promote-admin	Assigns full administrator privileges to a user
remove-license	Removes a license from a group
set-license	Sets a license and license functions for a group
show-deployment	Shows the current deployment
show-library-permissions	Shows permissions for a specific directory in the library
show-licenses	Shows licenses set on the server
switch-domain-name-style	Switches the domain names for all users and groups from one style (DNS or NetBIOS) to the other (for all configured domains)
update-deployment	Updates the current deployment

4.3.3.2 Authentication Commands

These commands are used to configure authentication.

config-auth	Configures authentication mode and default domain
config-auth-filter	Configures the Authentication Filter
config-basic-database-auth	Configures the Spotfire Database authentication source for use with the BASIC authentication method
config-basic-ldap-auth	Configures the LDAP authentication source for use with the BASIC authentication method
config-basic-windows-auth	Configures the Windows NT authentication source for use with the BASIC authentication method
config-client-cert-auth	Configures the CLIENT_CERT authentication method
config-external-auth	Configures the External authentication method
config-impersonation-auth	Configures the Impersonation authentication method
config-kerberos-auth	Configures the authentication service used with the Kerberos authentication method
config-ntlm-auth	Configures the authentication service used with the NTLM authentication method
config-post-auth-filter	Configures the Post Authentication Filter
config-two-factor-auth	Configures two-factor authentication
list-auth-mode	Displays the currently configured authentication mode
list-auth-config	Displays the current authentication configuration
list-ntlm-auth	Displays the NTLM authentication service configuration
list-post-auth-filter	Displays the current Post Authentication Filter configuration
set-auth-mode	Sets the authentication mode (deprecated, replaced by config-auth)
show-basic-ldap-auth	Shows the LDAP authentication source for use with the BASIC authentication method

4.3.3.3 Information Services Commands

These commands are used to configure Information Services.

add-ds-template	Adds a new data source template
clear-join-db	Clears the default join database configuration
create-join-db	Configures the default join database
export-ds-template	Exports the definition of a data source template
list-ds-template	Lists the data source templates
modify-ds-template	Modifies a data source template
remove-ds-template	Removes a data source template
show-join-database	Shows the configured default join database

4.3.3.4 JAAS Commands

These commands are used to manage JAAS configurations. The **test-jaas-config** command connects to the database in a read operation.

import-jaas-config	Imports new JAAS application configurations into the server configuration
list-jaas-config	Lists the JAAS application configurations
remove-jaas-config	Removes the specified JAAS application configurations from the server configuration
test-jaas-config	Tests a JAAS application configuration

4.3.3.5 Client Configuration Commands

These commands are used to configure clients connecting to Spotfire Server.

config-login-dialog	Configures the client login dialog behavior

4.3.3.6 Monitoring Commands

These commands are used to configure and administrate JMX access to the monitoring component. All monitoring commands connect directly to the database except for **config-jmx**.

config-action-log-database-logger	Configures the user action database logger
config-action-logger	Configures the user action logger.
config-action-log-web-service	Configures the action log web service
config-jmx	Configures the JMX RMI connector
create-jmx-user	Creates a new JMX user account
delete-jmx-user	Deletes a JMX user
list-jmx-users	Lists all JMX users

4.3.3.7 LDAP Commands

These commands are used to manage LDAP configurations for both authentication and the User Directory.

config-ldap-group-sync	Configures group synchronization for an LDAP configuration
create-ldap-config	Creates a new LDAP configuration to be used for authentication and/or the User Directory LDAP provider
list-ldap-config	Displays LDAP configurations
remove-ldap-config	Removes LDAP configurations
update-ldap-config	Updates LDAP configurations

4.3.3.8 Library Commands

These commands are used to configure the Spotfire Library.

check-external-library	Checks for inconsistencies between external storage and Spotfire database
config-import-export-directory	Configures the library import/export directory
config-library-external-data-storage	Configures the external library data storage

config-library-external-file-storage	Configures the file system storage of library item data
config-library-external-s3-storage	Configures the Amazon S3 storage of library item data
delete-library-content	Deletes library content
s3-download	Downloads the data of library items in Amazon S3 storage
show-import-export-directory	Shows the library import/export directory

4.3.3.9 Server Configuration Commands

These commands are used to perform basic server configuration tasks. Server configuration commands connect directly to the database except for **create-default-config**.

create-default-config	Creates a new server configuration file containing the default configuration
export-config	Exports a server configuration from the server database to the current working directory as a **configuration.xml** file
import-config	Imports a server configuration from a file to the server database
list-configs	Lists all available server configurations
set-config	Sets the current server configuration
set-restart-policy	Sets the server restart policy
show-config-history	Shows the configuration history
show-restart-policy	Shows the server restart policy

4.3.3.10 Server Database Commands

These commands are used to manage the server database connection pool. Server database commands connect directly to the database except for **bootstrap**, which can connect to the database to test the bootstrap configuration but does not change it.

bootstrap	Creates database connection information and stores it in the **bootstrap.xml** file. See “bootstrap.xml” on page 176.
modify-db-config	Modifies the common database connection configuration
set-db-config	Sets the common database connection configuration

4.3.3.11 User Directory Commands

These commands are used to configure the User Directory.

config-ldap-userdir	Configures the LDAP User Directory mode
config-userdir	Configures the User Directory
config-windows-userdir	Configures the Windows User Directory mode
list-ldap-userdir-config	Lists the configuration for the User Directory LDAP mode
list-userdir-mode	Lists the currently configured User Directory mode
list-userdir-config	Lists the current User Directory configuration
list-windows-userdir-config	Lists the configuration for the User Directory Window NT mode
set-userdir-mode	Sets the User Directory mode (deprecated, replaced by config-userdir)

4.3.3.12 Various Commands

config-attachment-manager	This command is used to configure the Attachment Manager which handles data transfer to and from Spotfire Server

4.3.3.13 Other Commands

help	Displays the help overview or a specific help topic
run	Runs a configuration script
version	Displays the current version of the server

4.4 Script Language

A script language is available. Create and run a script to invoke multiple commands in one go.

#

If a hash is the first character on a line, the line is a comment

Example: *# This is a comment that describes the next section.*

set

Defines a variable: The variable name and the value must be separated with an equal character (**=**)

Example: *set PASSWORD = "abc123"*

${Variable}

Substitutes the dollar sign and curly braces with the variable value:
If there is no matching variable, there is no substitution

Example: *--tool-password="${PASSWORD}"*

\

The logical line continues on the next line

Example: *bootstrap --no-prompt --driver-class="${DB_DRIVER}" \\
--database-url="${DB_URL}"*

echo

Writes to console

Example: *echo This message will be posted echo*

Empty rows are allowed

Note: Paths and comments using spaces must be enclosed in *straight* quotes (**"**). More advanced editors may change straight quotes to smart quotes, resulting in errors when running the commands.

4.4.1 Scripting a Configuration

You can create and run scripts of your own and two prepared scripts are provided in the **<installation dir>/tomcat/bin** folder. The **simple-config.txt** file sets up Spotfire Database authentication and User Directory and the **simple-config-ldap.txt** sets up LDAP authentication and User Directory.

Example: The **simple-config.txt** file is shown below, it is divided into three sections:

- The first lines describe how the script is executed using the run command
- Then a section of variables follows, variables that are used by the commands
- Finally the sections containing the commands follow

```
# Run this script from the command-line using the following command:
# config run simple-config.txt

# Before using this script you need to set the variables below:
set DB_DRIVER = "tibcosoftwareinc.jdbc.oracle.OracleDriver"
set DB_URL = "jdbc:tibcosoftwareinc:oracle://<server>:<port>;SID=\
<SID>"
#set DB_DRIVER = "tibcosoftwareinc.jdbc.sqlserver.SQLServerDriver"
#set DB_URL = "jdbc:tibcosoftwareinc:sqlserver://
<server>:<port>;DatabaseName=<database name>"
set DB_USER = "<db username>"
set DB_PASSWORD = "<db password>"
set CONFIG_TOOL_PASSWORD = "<config tool password>"
set ADMIN_USER = "<admin username>"
set ADMIN_PASSWORD = "<admin password>"

echo Creating the database connection configuration
bootstrap --no-prompt --driver-class="${DB_DRIVER}" --database-url=\
"${DB_URL}" \
  --username="${DB_USER}" --password="${DB_PASSWORD}"
--tool-password="${CONFIG_TOOL_PASSWORD}"
echo

echo Creating the default configuration
create-default-config
echo

echo Importing the configuration
import-config --tool-password="${CONFIG_TOOL_PASSWORD}" --comment=\
"First config"
echo

echo Creating the '${ADMIN_USER}' user to become administrator
create-user --tool-password="${CONFIG_TOOL_PASSWORD}" --username=\
"${ADMIN_USER}" --password="${ADMIN_PASSWORD}"
echo

echo Promoting the user '${ADMIN_USER}' to administrator
promote-admin --tool-password="${CONFIG_TOOL_PASSWORD}" --username=\
"${ADMIN_USER}"
echo
```

▶ Edit the script to make it work in your environment:

1 Open **simple-config.txt** in a text editor and edit the variables:

- If you use SQL Server, comment out the Oracle variables (“**#**”) and uncomment the SQL Server variables (remove “**#**”):
- For **DB_URL**, provide the specific values indicated by angle brackets.
- The **DB_USER** and **DB_PASSWORD** used in the **create_databases.bat** script (described in “Prepare the Database” on page 17). The **DB_USER** and **DB_PASSWORD** values are the Spotfire database user name and password.
- The **CONFIG_TOOL_PASSWORD**. Choose a Configuration Command Line Tool password that will be used to protect the server configuration from unauthorized access and/or modification.
- The **ADMIN_USER** and **ADMIN_PASSWORD**. First create the user, and then include the user in the group of Administrators (promote the user to the Administrator).

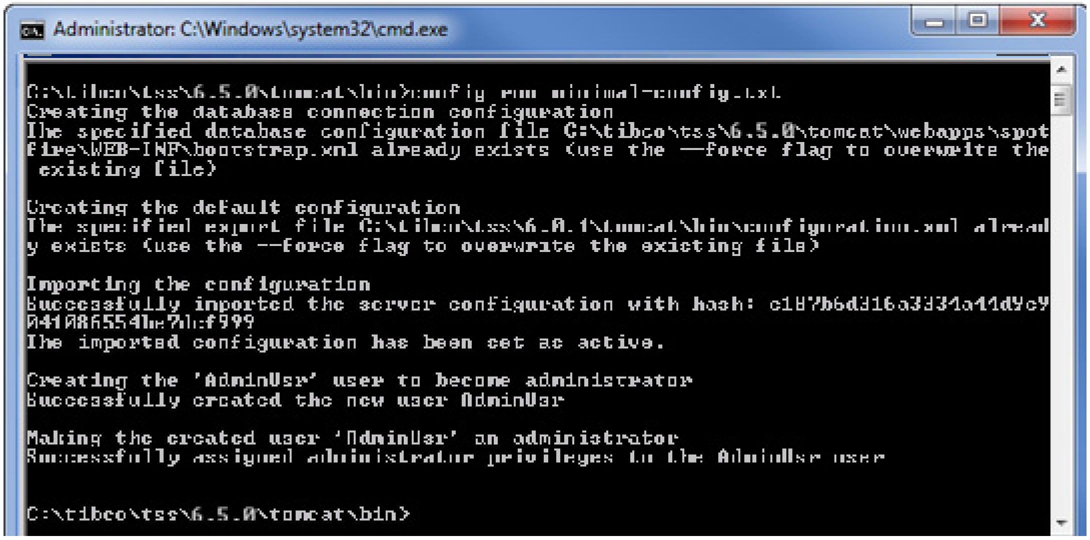
2 Save the script. If you do not want to overwrite the existing script, use another name.

▶ To run the script:

1 Start a command prompt and navigate to **<installation dir>\tomcat\bin**.

2 Type **config run simple-config.txt** and press **Enter**. The script executes and Spotfire Server receives a basic configuration. that the tool provides feedback when running the script. In this case a script has already been run:

- The tool is conservative and does not overwrite the **bootstrap.xml** or **configuration.xml** located in the **<installation dir>/tomcat/bin** unless the **--force** flag is used.
- The **AdminUser** is created and promoted to Administrator: This user did not exist.



Administrator: C:\Windows\system32\cmd.exe

```
C:\tibco\tss\6.5.0\tomcat\bin>config run minimal-config.txt
Creating the database connection configuration
The specified database configuration file C:\tibco\tss\6.5.0\tomcat\webapps\spot
fire\WEB-INF\bootstrap.xml already exists (use the --force flag to overwrite the
 existing file)

Creating the default configuration
The specified export file C:\tibco\tss\6.0.1\tomcat\bin\configuration.xml alread
y exists (use the --force flag to overwrite the existing file)

Importing the configuration
Successfully imported the server configuration with hash: c187b6d316a3334a11d9c9
041086554be7bcf999
The imported configuration has been set as active.

Creating the 'AdminUsr' user to become administrator
Successfully created the new user AdminUsr

Making the created user 'AdminUsr' an administrator
Successfully assigned administrator privileges to the AdminUsr user


C:\tibco\tss\6.5.0\tomcat\bin>
```

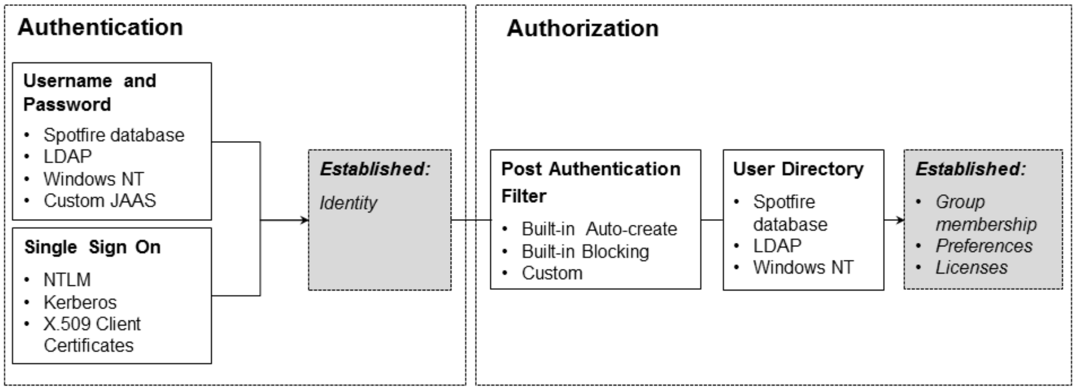
3 It is recommended to manually remove the **configuration.xml** when you are done.

Note: Do not remove **bootstrap.xml**. It is required to start and run the server. See “bootstrap.xml” on page 176.

Note: The **simple-config.txt** file contains sensitive information.

4 To close the command prompt, type **exit** and press **Enter** or close the window.

4.5 Authentication and User Directory



4.5.1 Authentication Methods

Spotfire Server supports several authentication methods:

- Authentication methods based on user names and password are described in “User Name and Password Authentication Methods” on page 57.
- Single sign-on authentication methods like NTLM, Kerberos and X.509 Client Certificates are described in “Single Sign-On Authentication Methods” on page 61.

4.5.2 User Directory

Spotfire Server stores the names of the users, and optionally also their passwords, in its User Directory. In the User Directory, it is possible to organize the users in groups. The user and group information can later be used to assign permissions, licenses, preferences etc. to the different resources available within the Spotfire system.

To integrate existing IT systems, the User Directory can use external user-handling systems. Administrators do not have to replicate user lists or group hierarchies from their existing environments in Spotfire Server. The external directory is usually an LDAP directory, such as Windows AD, but Spotfire Server also offers a legacy Windows NT Domain integration, where the user information is collected from Windows NT domain controllers.

Starting with version 5.0, Spotfire Server only accesses the external directories during its periodical synchronization of the User Directory, though password verification will require a connection, as well as checking for new users. Spotfire Server will periodically refresh the information about the users and the groups with the current information from the external directories.

For a comprehensive list of user directory modes, and how they can be combined with different Authentication methods, See “User Directory Modes” on page 80.

4.5.3 Authentication and Post-Authentication Filter

When users log in to Spotfire Server, their identities are verified in the following steps:

1 The credentials are validated in order for Spotfire Server to confirm the identity of the user. Typical credentials are user names and passwords, NTLM tokens, Kerberos tickets or X.509 client certificates.

- If Spotfire Server is configured to store both user names and hashed passwords in its database, the server can validate the user-supplied credentials by itself.
- If Spotfire Server is set up to integrate with an external directory, the validation responsibility is delegated to this system. Such a system can for instance be an LDAP directory or a Windows domain controller.

2 When the user identity is established, an extra post-authentication check is performed by Spotfire Server's Post-Authentication Filter. The filter has two built-in modes: blocking and auto-creating:

- *Blocking mode:* The filter simply blocks all users that are not already present in the server's User Directory.
- *Auto-creating mode:* The filter automatically creates new accounts for any user that logs into the server for the first time. This mode is only valid if Spotfire database mode is configured.

The blocking mode is the default mode. When it is used with a User Directory in LDAP/Active Directory mode, it automatically transforms to the domain name of the authenticated user to match the configured domain name style.

The auto-creating mode is typically applied when using an LDAP directory or X.509 certificates for authentication together with the User Directory set up in database mode. The Post-Authentication Filter will create users with their external domain names, even though the User Directory is in database mode, unless the **collapse domains** configuration property is enabled. This makes it possible to later switch to LDAP or Windows NT mode. If the **collapse domains** configuration property is enabled, the users will be created within the internal SPOTFIRE domain and it will not be possible to later switch to LDAP or Windows NT mode.

It is also possible to use Spotfire Server's API to create a custom Post-Authentication Filter to perform additional validation. This filter must be installed in the **<installation dir>/tomcat/webapps/spotfire/WEB-INF/lib** directory on all servers. It is enabled using the “config-post-auth-filter” on page 224. If a custom filter is used, it will be combined with the built-in filter, meaning that the filters will work together. This is a change from previous releases where the custom filter had to be used instead of a built-in filter.

3 When the user identity is both confirmed (using the credentials) and filtered (meaning that the user exists in the User Directory), an authority check is made to decide whether or not the user is allowed to access the requested resource and what the user can do with regard to licenses etc.

4.6 User Name and Password Authentication Methods

When users start a Spotfire client, they are presented with a login dialog where they select which Spotfire Server to connect to. If that server uses a user name and password based authentication method, the users are also prompted for user name and password.

The user name and password are then sent to Spotfire Server (over the HTTP BASIC protocol). The user name and password authentication methods are sometimes referred to as BASIC authentication methods. The credentials are not encrypted then they are transferred unless the server uses TLS/SSL. and the information can easily be collected by other eavesdropping computers on the network. To use any user name and password authentication method in a safe manner, make sure to also enable TLS/SSL to safely transfer the user name and the password to Spotfire Server over the encrypting HTTPS protocol.

The user name and password can be validated using:

- Spotfire database
- LDAP Directory (for example Active Directory)
- Windows NT Domain (Legacy, use only if you cannot use LDAP)
- Custom JAAS

For all methods, entries are created in the Spotfire database. When using an external authentication method, appropriate information is copied to the Spotfire database.

4.6.1 Authentication towards the Spotfire Database

This authentication method requires that the User Directory is configured for Spotfire database. The database will store the names and password hashes of all users, and an administrator will have to create all user accounts in advance. This is the default behavior, and no configuration is needed for this authentication method. This is a configuration that is easy and fast to set up and it is recommended for small sites.

To create a lot of users at once, export the users from an external system and imported to the Spotfire Database using the Administration Manager.

4.6.2 Authentication towards LDAP

This authentication method integrates with an existing LDAP directory and delegates the actual authentication responsibility to its configured LDAP servers. The result is that only users with valid accounts in the LDAP directory can log in to Spotfire Server. This setup is recommended for all larger sites. It can be combined with both Spotfire database User Directory and LDAP User Directory.

It is recommended to combine the LDAP authentication method with an LDAP User Directory mode. However, in some cases, for example where the LDAP directory contains a very large number of users that are not divided into convenient sub-units (contexts), combining the LDAP authentication method with a Spotfire database User Directory will reduce the set of users tracked within Spotfire Server. Only the users that are logging in to Spotfire Server will be included. This makes Spotfire Server's User Directory easier to manage and survey.

When combining it with a Spotfire database User Directory configuration, the users shall be automatically added to the User Directory and consequently the Post-Authentication Filter must be configured in auto-creating mode. When combining it with an LDAP User Directory mode, the default setting of the Post-Authentication Filter, blocking mode, is already correct.

Spotfire Server supports the following LDAP servers:

- Microsoft Active Directory
- The Directory Server product family (Oracle Directory Server, Sun Java System Directory Server, Sun ONE Directory Server, iPlanet Directory Server, Netscape Directory Server)

The above mentioned are the tested and supported variants. Other types of LDAP servers may also work with Spotfire Server. Such a custom LDAP configuration may be slightly more advanced to configure.

Note: When Spotfire Server is authenticating towards a Microsoft Active Directory server, it will automatically use the Fast Bind Control (also known as Concurrent Bind Control) option to minimize the consumed resources on the LDAP server.

4.6.3 Configuring SASL Authentication for LDAP

DIGEST-MD5 and GSSAPI are SASL (Simple Authentication Socket Layer) mechanisms. These are used for secure authentication of Spotfire Server when it is connecting to LDAP Servers. SASL prevent clear text passwords from being transmitted over the network.

4.6.3.1 DIGEST-MD5

When configuring SASL authentication with DIGEST-MD5 in an Active Directory environment, the distinguished name (DN) does not work for authentication and the userPrincipalName attribute must be used instead. The **authentication attribute** option should be set to userPrincipalName and the **username attribute** option should be set to sAMAccountName, which is the default value for an Active Directory LDAP configuration.

When setting up SASL with DIGEST-MD5 in an Active Directory environment, all accounts must use reversible encryption for their passwords. This is typically not the default setting for Active Directory.

4.6.3.2 GSSAPI

These instructions apply for Active Directory LDAP configurations. Spotfire Server does not support GSSAPI for other LDAP alternatives.

The GSSAPI authentication mechanism provides secure authentication even over un-secure networks since it uses the Kerberos protocol for authentication. Passwords are not sent in clear text across the network even if using un-encrypted HTTP, for information about Kerberos in general, see “Kerberos Authentication” on page 66.

▶ Configure Spotfire Server for GSSAPI Authentication of LDAP

Preparations:

1 Make sure you have a fully working Active Directory LDAP configuration using clear-text password authentication (also known as **simple** authentication mechanism). This configuration is created using the Configuration Tool or the Configuration Command Line Tool.

- Save this fully working Active Directory LDAP configuration to file.
- the LDAP configuration's ID.

2 Make sure that you have a fully working **krb5.conf** file. The content of the **krb5.conf** file shall be the same as when setting up Spotfire Server for Kerberos authentication. See “Configure Kerberos for Java:” on page 70.

Note: Make sure to stop the entire service/Java process before installing the file. It is not sufficient that the restart-policy is set automatic force or automatic on idle. If the **krb5.conf** file is modified after Spotfire Server has been started, a restart of Spotfire Server process is required for the modifications to have effect.

Procedure:

1 Stop Spotfire Server, See “Start and Stop Spotfire Server” on page 110.

2 Copy the fully working **krb5.conf** file to the **<inst dir>/jdk/jre/lib/security** directory on each Spotfire Server in the cluster.

3 Start the Configuration Tool and provide the Tool Password, See “Configuration Tool” on page 34.

4 Go to the LDAP *Configuration* Panel.

5 Update the LDAP user name so that it is a proper Kerberos principal name. Usually it is sufficient to add the name of the account's Windows domain written in upper-case letters. Sometimes its also necessary to include the Windows domain name as well. Using a name based on a distinguished name (DN) or including a NetBIOS domain name does not work when using GSSAPI. Examples of correct names: "ldapsvc@ RESEARCH.EXAMPLE.COM" and "ldapsvc@research.example.com@ RESEARCH.EXAMPLE.COM".

6 Select the specific LDAP configuration to be GSSAPI enabled and expand the **Advanced settings.**

7 Set the **security-authentication** configuration property to **GSSAPI**.

8 Set the authentication-attribute to **sAMAccountName** or **userPrincipalName** (select what works best for your configuration). The default value is empty.

Note: If the **krb5.conf** file contains more than one Kerberos realm, the authentication-attribute must be set to "userPrincipalName".

9 Add a custom property with the key kerberos.login.context.name and the value **SpotfireGSSAPI**.

10 Save the configuration to the Spotfire database by clicking **Save configuration**.

11 Start Spotfire Server, See “Start and Stop Spotfire Server” on page 110.

Procedure steps related to LDAP configurations need to be performed for each LDAP catalogue that shall have GSSAPI enabled. For multiple LDAP configurations, repeat these steps for each configuration.

4.6.4 Authentication towards Windows NT Domain (legacy)

With this authentication method, user authentication is delegated to Windows NT domain controllers. To be able to use this method, Spotfire Server must be installed on a machine running Windows and you must have a working Windows NT 4 Server Domain Controller or a Windows Server 2000 (or later) Domain Controller running in Mixed Mode. This is a legacy solution that should only be used if LDAP cannot be used.

Just like the LDAP authentication method, the Windows NT Domain authentication method can be combined with a User Directory in either Windows NT Domain mode or in Spotfire database mode.

When combining this authentication method with a Spotfire database User Directory mode, the Post-Authentication Filter must be configured in auto-creating mode, so that the users will be automatically added to the User Directory. When combining it with a Windows NT Domain User Directory mode, the default blocking Post-Authentication Filter is already correct.

4.6.5 Authentication towards a Custom JAAS Module

All authentication methods described above are implemented as Java Authentication and Authorization Service (JAAS) modules. Spotfire also supports third-party JAAS modules. You may therefore use a custom JAAS module, provided that it validates user name and password authentication and that it uses JAAS’ **NameCallback** and **PasswordCallback** objects for collecting the user names and passwords.

When using a custom JAAS module, you must place the **jar** file in the **<installation dir>/tomcat/webapps/spotfire/WEB-INF/lib** directory on all Spotfire Servers.

Consult the JAAS Reference Guide for more information about JAAS.

4.7 Single Sign-On Authentication Methods

Spotfire Server is capable of integrating with some single sign-on systems used in enterprise environments. A single sign-on system can be defined as an authentication system that provides access to many resources once the user is initially authenticated.

Spotfire Server can use the NTLM or Kerberos single sign-on authentication methods, where the identity information stored within the user's current Windows session is reused to authenticate the user on the server. Thus, when using these authentication methods, the user is never prompted for user name or password when logging in to Spotfire Server. The Kerberos and NTLM authentication methods are commonly referred to as Windows Integrated Authentication.

Spotfire Server can also authenticate users based on X.509 certificates. This requires the server to be configured for mutual SSL, meaning HTTPS with X.509 client certificates.

4.7.1 NTLM Authentication

The NTLM authentication method reuses the identity information associated with the user's current Windows session that is created when the user initially logs in to Windows. When both the client computer and the server computer belong to the same Windows domain or two separate Windows domains with established trust between them, this can provide a single sign-on experience.

If the client computer belongs to a separate Windows domain (without trust established to the server computer's domain), the current Windows session is not valid in the Windows domain of the server computer and the user will be prompted for user name and password. The user must then enter user name and password of a valid account that belongs to the Windows domain of the server computer.

It is not possible to delegate NTLM authentication, Spotfire Server can not reuse the authentication credentials presented by the client, for example when authenticating against an Information Services data source that also uses NTLM. If you need such functionality, you must use Kerberos instead.

Upgrading to 5.0 or later: Spotfire Server supports NTLMv2 since version 3.2. The older NTLMv1 authentication mechanism was deprecated in version 4.5 and has now been removed. The instructions below explain how to set up the newer NTLMv2 authentication mechanism.

The NTLM authentication method needs to be combined with a User Directory in either:

- LDAP mode, recommended, see “User Directory in LDAP Mode” on page 81.
- Spotfire database mode, provided that the default Post-Authentication Filter is configured in auto-creating mode, see “User Directory in Spotfire Database Mode” on page 80.

The following instructions assume that either of these combinations is already fully working.

When using the NTLM authentication method, the User Directory is typically configured for the NetBIOS domain name style.

Setting up NTLM authentication involves two steps:

1 Creating a computer service account in your Windows Domain

You must create a computer service account in your Windows Domain. A Visual Basic script, **SetupWizard.vbs** (developed by IOPLEX Software) is distributed with Spotfire Server and will perform this task. The script must be run on a Windows machine, but does not have to be run on the same machine as the server is installed on.

If you are unable to run this script, or prefer to create the account manually, *make sure to create a computer account*. A user account will not work. Reusing an existing computer account will not work. See “Creating a Computer Account Manually” on page 63.

2 Configuring NTLM authentication using configuration commands.

Note: If you have more than one Spotfire Server in the cluster, you must also perform additional steps on each Spotfire Server.

▶ To create a computer service account in your Windows domain:

You must be logged into your Windows domain as a member of the group Account Operators or Administrators to run the **SetupWizard.vbs** script.

1 Double click on the setupwizard.vbs script located in the directory **<installation dir>/tomcat/bin**. If the server is installed on a Linux or Solaris machine, the script has to be copied to a Windows machine first.

2 In the *Domain Controller Hostname* panel, enter the hostname of one of your domain controllers. Click **OK**.

3 In the Account Name panel, enter the short name of the computer account to be created. The short name must not exceed 15 characters. Click **OK**.

4 In the *Distinguished Name* panel, enter a distinguished name for the account to be created. A distinguished name based on the short name entered in the previous panel is suggested. You should edit this to match your Windows domain, with regards to parameters such as in which Organizational Units (OU) the account should be placed. Click **OK**.

5 In the *Account Password* panel, enter a password for the account to be created. Click **OK**.

6 A dialog will show with text indicating if the tool was successful or not. Click **OK**.

Note: If the tool was unsuccessful, you should make sure that the logged in user has the required permissions to create accounts in the Windows Domain, and that the Domain Controller can be reached.

7 The file **SetupWizard.txt**, created by the tool in the folder where the tool is located, will open. If it does not, open it manually. The information is required to run the NTLM authentication configuration commands. File example:

```
# Generated by the Jespa Setup Wizard from IOPLEX Software on 2011-04-07

jespa.bindstr = dc.example.research.com
jespa.dns.servers = 192.168.0.1
jespa.dns.site = Default-First-Site-Name
jespa.service.acctname = jespa-svc$@dc.example.research.com
jespa.service.password = Pa33w0rd
```

Creating a Computer Account Manually

If you prefer to create the computer account manually, you should do so using the Microsoft Management Console snap-in Domain Users and Computers. Refer to Microsoft documentation for details on how to use this tool.

When you have created a new computer account, you need to set a password for this account. Unfortunately, this is not possible to do in the Microsoft Management Console. In the directory **<installation dir>/tomcat/bin** there is a VBS script called **SetComputerPassword.vbs**. Run this script from the command line with the account name and password as arguments to the command.

Note: The **SetComputerPassword.vbs** file can only be executed on a Windows machine. The script must be copied to a Windows machine, but does not have to be run on the same machine as the server is installed on.

Example:

SetComputerPassword.vbs jespa-svc$@dc.example.research.com Pa33w0rd

▸ To configure NTLM for a single server

Use the Configuration tool

or

1 Use the command **config-ntlm-auth** (page 221) and **list-ntlm-auth** (page 261) to configure NTLM authentication.

2 Use the **set-auth-mode** (page 273), import the configuration and restart the server to activate the NTLM SSO authentication method.

To run these commands, you need some of the specific information described below.

Server (optional)	The name of the server instance to which the specified configuration options belong. If no server name is specified, then all parameters will be shared, applying to all servers in the cluster. It is common to use server-specific values for the **account name**, **password** and **localhost NetBIOS name** configuration options.

Account name
(required)

Specifies the fully qualified name of the Active Directory computer account that is to be used by the NTLM authentication service. This account must be a proper computer account, created solely for the purpose of running the NTLM authentication service. It can neither be an ordinary user account, nor an account of an existing computer. that the local part of an Active Directory computer account name always ends with a dollar sign, for instance: **ntlm-svc$@research.example.com**. The local part of the account name (excluding the dollar sign) must not exceed 15 characters.

Example: **ntlm-svc$@research.example.com**

Password
(required)

Specifies the password for the computer account used by the NTLM authentication service.

DNS domain name
(optional)

The DNS name of the Windows domain to which Spotfire Server's computer belongs. The specified domain name will automatically be resolved into domain controller hostnames. As an alternative to specifying a DNS domain name, it is also possible to specify a domain controller hostname directly. It is recommended to use the DNS domain name option, since you then automatically get the benefits of fail-over and load-balancing, provided that you have more than one domain controller. The DNS domain name and domain controller arguments are mutually exclusive.

Example: **research.example.com**

Domain controller
(optional)

The DNS hostname of an Active Directory domain controller. It is recommended that the DNS domain name option is to be used instead, since that option gives the benefits of fail-over and load-balancing. The domain controller and DNS domain name arguments are mutually exclusive.

Example: **dc01.research.example.com**

DNS servers
(optional)

A comma-separated list of IP addresses of the DNS servers associated with the Windows domain. When no DNS servers are specified, the server will fall back to use the server computer's default DNS server configuration.

Example: **192.168.1.1,192.168.1.2**

AD site
(optional)

Specifies the Active Directory site where the Spotfire system is located. Specifying an Active Directory site can potentially increase performance, since the NTLM authentication service will then only communicate with the local Windows domain controllers.

Example: **VIENNA**

DNS cache TTL
(optional)

Specifies how long (in milliseconds) name server lookups should be cached. The default value is **5000** ms.

Localhost NetBIOS name (optional)
Specifies the NetBIOS name, used by a server to identify its connection to the domain controller. The default value is derived from the account name option. This option is only necessary to specify when there is more than one server in the cluster. Since a domain controller only allows one connection per NetBIOS name, a cluster with multiple servers must either use separate NTLM accounts for each server or explicitly specify unique localhost NetBIOS names for the servers. The localhost NETBIOS name must not exceed 15 characters in length.

Example: **ntlm-svc-server1** (for **server1.research.example.com**)

Connection ID header name (optional)
This parameter specifies the name of an HTTP header containing unique connection IDs in environments where the server is located behind some kind of proxy or load-balancer that does not properly provide the server with the client's IP address. The specified HTTP header must contain unique connection IDs for each client connection and is thus typically based on the client's IP address together with the connection's port number on the client side.

To configure NTLM for a cluster with multiple servers

To set up NTLM for a cluster with multiple servers, start with configuring the options common to all servers in the cluster. This is performed according to the instructions in "To configure NTLM for a single server" on page 63, with the following modifications.

This step involves specifying a **DNS domain name** (recommended) or a **domain controller** (not recommended) and possibly also an **AD site** name. The **account name** and **password** options must be left out at this point (will be specified later). It is also very important that the **server** argument is not specified at this stage.

The common NTLM configuration now needs to be completed with account information for each Spotfire Server in the cluster. When a server logs in to the domain controller, its identity is based on the name of the computer account it uses for the connection. The resulting name is known as a localhost NetBIOS name. Since a domain controller only allows one connection per localhost NetBIOS name, multiple servers typically cannot login using the same computer account. Thus, each server ideally uses its own NTLM account.

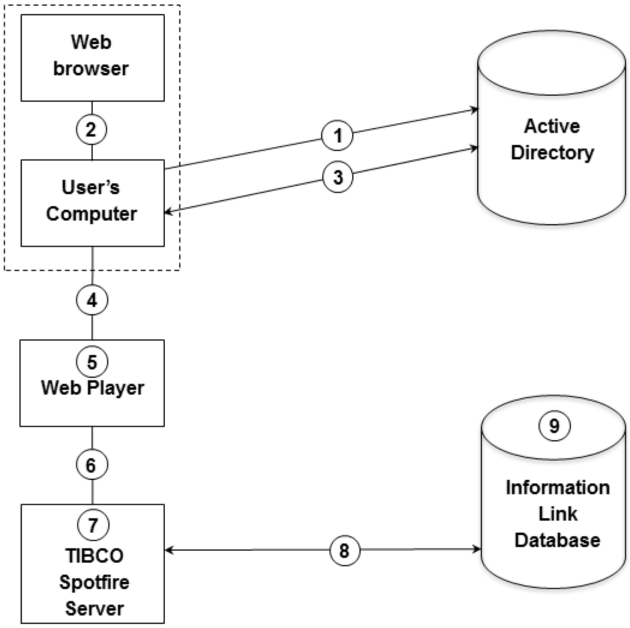
Note: Sometimes, like when running two servers on the same computer, it happens to be possible to actually share the NTLM account by explicitly specifying individual localhost NetBIOS names that are used instead of the name derived from the NTLM account.

- If separate NTLM accounts are to be used, then use the **account name** and **password** options to specify the server's own NTLM account.
- If a shared NTLM account is to be used, specify the **account name** and **password** for the shared account, as well as a unique **localhost NetBIOS name**. The localhost NetBIOS names must not exceed 15 characters.

When the decision has been made whether to use individual NTLM accounts or share an NTLM account by explicitly specifying localhost NetBIOS names, the command

config-ntlm-auth is run again, once for each server in the cluster. The command will update Spotfire Server configuration with the cluster server's specific configuration options. This time, the **server** argument must be specified so that it reflects the server name, as defined in the server's **bootstrap.xml** file.

4.7.2 Kerberos Authentication



1 The user logs in to Windows.

2 The user's computer tries to connect to the Spotfire Web player.

3 The user's computer receives a Kerberos ticket from the Active Directory to connect to Spotfire Web player.

4 The user's computer forwards the Kerberos ticket to the Spotfire Web player.

5 The Spotfire Web Player validates the received Kerberos ticket.

6 The Spotfire Web Player forwards the Kerberos ticket to the Spotfire Server.

7 The Spotfire Server validates the Kerberos ticket.

8 When executing an Information Link, the Spotfire Server forwards the Kerberos ticket to the Information Link database.

9 The information link database validates the Kerberos ticket.

The picture illustrates a simplified overview of Kerberos Authentication with delegation.

Kerberos is a single sign-on protocol that allows for secure authentication even over unsecure networks. The Kerberos protocol uses tickets for authentication instead of user names and passwords. The tickets are issued by a centralized Kerberos server and contains information that only the intended target of the ticket can decrypt. In Microsoft Windows environments, the domain controllers act as Kerberos servers, and every user automatically signs in to Kerberos when logging in to the Windows desktop. Kerberos can be a bit hard to set up, but once it is fully working you have a very secure authentication system with the benefits of single sign-on.

Prerequisites

- Windows Domain Controllers running Windows Server 2003 SP1 or later.
- A computer with the **Microsoft Active Directory Users and Computers** MMC snap-in.
- A computer with the **Microsoft Support Tools** installed.
- A domain administrator account or a user account which is a member of the built in **Account Operators** domain group, or any account with similar permissions.
- Windows Domain accounts for all Spotfire users.

- A fully working User Directory in place, either in
  - LDAP mode (recommended) or
  - Spotfire database mode, provided that the built-in Post-Authentication Filter is **auto-creating**

It us usually a good idea to first create a working setup where the server uses Basic/ LDAP authentication and a User Directory in LDAP mode and then proceed with switching from Basic/LDAP to Kerberos.

4.7.2.1 Configuration Instructions

The following instructions are required to configure Spotfire Server for the Kerberos authentication method.

As a Domain Administrator:

1 **Create a Kerberos service account:**

In this step the Kerberos service account is created. The following examples will assume that the account's name is **spotsvc**.

Logged in as a domain administrator or a user which is a member of the built in **Account Operators** domain group, launch the **Active Directory Users and Computers** MMC snap-in and create a normal user account with the following properties:

- Use the same identifier in the **Full name** and **User logon name (pre-Windows 2000)** fields and make sure to use only lower case characters and that there are no spaces in these fields.
- Select the **Password never expires** option.
- Clear the **User must change password at next logon** option.
- If Kerberos unconstrained delegation is to be used for Information Services data sources, the account option **Account is trusted for delegation** must also be selected.
- Kerberos constrained delegation can also be used for Information Services data sources, but is set up on a service-by-service basis and is not described here.

2 **Register Service Principal Names:**

While still logged in as a domain administrator or as a user which is a member of the built in **Account Operators** domain group, use the **setspn.exe** command-line tool to register two Service Principal Names (SPNs) for the Kerberos service account. The setspn.exe command-line tool is a part of the Microsoft Support Tools package which is typically installed on domain controllers. The Support Tools can also be downloaded from Microsoft's web page.

The **setspn.exe** tool for Windows Server 2008 or later has been improved with extra argument checking to prevent that no duplicate Service Principal Names are created. If you use the improved version of the setspn.exe tool, then execute the following two commands to register the Service Principal Names:

```
> setspn -S HTTP/<fully qualified hostname>[:<port>] <service account name>
> setspn -S HTTP/<hostname>[:<port>] <service account name>
```

If you are using the **setspn.exe** tool for Windows Server 2003 or earlier, the extra argument checking is not supported. Instead, execute the following two commands to register the Service Principal Names:

```
> setspn -A HTTP/<fully qualified hostname>[:<port>] <service account name>
> setspn -A HTTP/<hostname>[:<port>] <service account name>
```

Note: It is recommended not to have multiple Kerberos-enabled HTTP services on one machine.

Replace the **<fully qualified hostname>**, **<service account name>**, **<hostname>** and **<port>** with the appropriate values. **Note:** It is vital to note that all values are case sensitive.

- **fully qualified hostname**: The fully qualified DNS hostname of the computer hosting Spotfire Server (written in lower case)
- **hostname**: The short DNS hostname, without domain suffix, of the computer hosting Spotfire Server (written in lower case)
- **service account name**: The **user login name** of the previously created Kerberos service account (written in lower case)
- **port**: The TCP port number that Spotfire Server is listening on

Note: You must use the name of an **A record** for Spotfire Server. A **CNAME** record will not work.

Note: Avoid explicitly specifying the port number if Spotfire Server is using the default HTTP port 80.

Example: Registering Service Principal Names for the **spotsvc** Kerberos service account to be used by a Spotfire Server installed on the **spotfireserver.research.example .com** computer and listening on the default HTTP port 80 or the default HTTPS port 443:

```
> setspn -A HTTP/spotfireserver.research.example.com spotsvc
> setspn -A HTTP/spotfireserver spotsvc
```

This will create these two Service Principal Names:

- **HTTP/spotfireserver.research.example.com**
- **HTTP/spotfireserver**

Example: Registering Service Principal Names for the **spotsvc** Kerberos service account to be used by a Spotfire Server installed on the **spotfireserver.research.example .com** computer and listening on the non-default HTTP port 8080:

```
> setspn -A HTTP/spotfireserver.research.example.com:8080 spotsvc
> setspn -A HTTP/spotfireserver:8080 spotsvc
```

This will create two SPNs:

- **HTTP/spotfireserver.research.example.com:8080**
- **HTTP/spotfireserver:8080**

To list the resulting Service Principal Names for a Kerberos service account, you can execute the following command:

```
> setspn -L <service account name>
```

Example: Verifying Service Principal Names for the **spotsvc** Kerberos service account

```
> setspn -L spotsvc
```

3 Create a keytab file for the Kerberos service account:

While still logged in as a domain administrator or as a user which is a member of the built in **Account Operators** domain group, execute the following command:

```
> ktpass /princ HTTP/<fully qualified hostname> [:<port>]@<realm> /ptype
krb5_nt_principal /crypto rc4-hmac-nt /mapuser <service account name> /out
spotfire.keytab -kvno 0 /pass *
```

Replace the **<fully qualified hostname>**, **<port>**, **<realm>**, and **<service account name>** with the appropriate values.

Note: It is vital to note that all values are case sensitive.

- **fully qualified hostname**: The fully qualified DNS hostname of the computer hosting Spotfire Server, which must exactly match the fully qualified hostname used when registering the SPNs (written in lower case)
- **port**: The TCP port number that Spotfire Server is listening on (only specified if the port number was explicitly included in the registered SPNs)
- **realm**: The name of the Kerberos realm, which is the DNS domain name written in upper case
- **service account name**: The **user login name** of the service account with the registered SPNs (written in lower case)

The tool will prompt for the password of the service account. Enter the same password as when creating the service account.

It is not critical to use the name **spotfire.keytab** for the keytab file. However, the remaining instructions will assume that this is the name of the keytab file.

Note: If you ever change the password of the Kerberos service account in the future, you must re-create the keytab file.

Note: Older versions of the **ktpass.exe** tool will fail to create the keytab file when it is not being run on an actual domain controller.

Example: Creating a keytab file for the **spotsvc** Kerberos service account in the **research.example.com** domain for Spotfire Server listening on the default HTTP port 80 on the **spotserver.research.example.com** computer:

```
> ktpass /princ HTTP/spotfireserver.research.example.com@RESEARCH.EXAMPLE.COM /
ptype krb5_nt_principal /crypto rc4-hmac-nt /mapuser spotsvc /out spotfire.keytab -kvno
0 /pass *
```

Example: Creating a keytab file for the **spotsvc** Kerberos service account in the **research.example.com** domain for Spotfire Server listening on the HTTP port 8080 on the **spotserver.research.example.com** computer:

```
> ktpass /princ HTTP/spotfireserver.research.example.com:8080@
RESEARCH.EXAMPLE.COM /ptype krb5_nt_principal /crypto rc4-hmac-nt /mapuser
spotsvc /out spotfire.keytab -kvno 0 /pass *
```

On Spotfire Server:

4 **Copy the Kerberos service account's keytab file to Spotfire Server:**

Copy the **spotfire.keytab** file to the directory **<installation dir>\jdk\jre\lib\security** (Windows) or **<installation dir>/jdk/jre/lib/security** (Unix) on Spotfire Server.

Note: Since this file contains sensitive information it must be handled with care. The file must not be readable for unauthorized users.

To list the contents of the keytab file, use the **klist** command-line tool which will list the principal name and security credentials. The tool is included in the bundled JDK and is only available when installed on Windows:

```
> <installation dir>\jdk\jre\bin\klist.exe -k -t -K <keytab file>
```

To test the keytab file, use the **kinit** command-line tool which is also included in the bundled JDK on Windows platforms:

```
> <installation dir>\jdk\jre\bin\kinit.exe -k -t < keytab file> HTTP/<fully qualified
hostname> [:<port>]@<realm>
```

If the keytab file is correctly set up, a ticket cache file will be created in the logged in user's home directory. It can typically be found with the path **C:\Users\<user>\krb5cc_<user>**. As soon as you have verified that the ticket cache was created, you must delete the ticket cache file to prevent future problems.

5 **Configure Kerberos for Java:**

Open the file **krb5.conf** located in the directory **<installation dir>\jdk\jre\lib\security** (Windows) or **<installation dir>/jdk/jre/lib/security** (Unix) and edit the following values to reflect your environment:

- **MYDOMAIN**: The name of the Kerberos realm, usually the same as the name of the Windows Domain, written in upper case
- **mydomain**: The name of the Windows Domain, written in lower case
- **mydc**: The name of the domain controller, written in lower case

Note: The arguments are case-sensitive. It is critical to use the correct case for these values!

For more information, See "krb5.conf" on page 177.

Example: Configuring Kerberos for Java in the **research.example.com** domain, with the two domain controllers **dc01.research.example.com** and **dc02.research.example.com**:

```
===============
Krb5.conf
===============
[libdefaults]
default_realm = RESEARCH.EXAMPLE.COM
default_keytab_name = spotfire.keytab
default_tkt_enctypes = rc4-hmac
default_tgs_enctypes = rc4-hmac
[realms]
RESEARCH.EXAMPLE.COM = {
kdc = dc01.research.example.com
kdc = dc02.research.example.com
admin_server = dc01.research.example.com
default_domain = research.example.com
}
[domain_realm]
.research.example.com = RESEARCH.EXAMPLE.COM
research.example.com = RESEARCH.EXAMPLE.COM
[appdefaults]
autologin = true
forward = true
forwardable = true
encrypt = true
```

6 **Select Kerberos as the Spotfire login method:**

Use the Configuration Tool

or

- Use the **config-kerberos-auth** command (page 213) to configure the Kerberos authentication method. The command takes the following two parameters:
    - **Keytab file**: The fully qualified path to the **spotfire.keytab** file. If the keytab file is named **spotfire.keytab** and has been copied to the recommended directory, the default path **${java.home}/lib/security/spotfire.keytab** is already correct. The shorthand **${java.home}** refers to the directory **<installation dir>\jdk\jre** (Windows) or the **<installation dir>/jdk/jre** (Unix)
    - **Service Principal Name**: Specify the same Service Principal Name that was used when creating the keytab file. *Example*: **HTTP/spotfireserver.research.example.com**
- Use the **set-auth-mode** command (page 273) to activate the Kerberos SSO authentication method.
- Import the configuration and restart the server for the changes to have effect.

7 **Disable user name and Password fields in client login dialog:**

Since the Kerberos authentication method provides single sign-on capabilities, there is no need to prompt an end user for user name and password in the Spotfire client login dialog. In fact, any entered user name and password is unlikely to work, even if the credentials are fully valid.

Use the Configuration Tool or the following **config-login-dialog** command to disable the user name and password fields in the Spotfire client login dialog:

```
> config  config-login-dialog  --allow-user-provided-credentials=false
```

(For more information about the **config-login-dialog** command, go to page 220.)

Note: If you are using the Configuration Tool, select **Never display login dialog** for the **Login dialog** option**.**

Then, import the new configuration and restart the server.

4.7.3 Authentication Using X.509 Client Certificates

When Spotfire Server is set up with HTTPS and is set to require client certificates, the information from the certificates can also be used for login purposes.

This authentication method authenticates users using an X.509 Client Certificate from the Spotfire client to Spotfire Server.

A prerequisite for this authentication method is that Spotfire Server is set up with HTTPS and is set to require client certificates.

Perform the following steps to configure Spotfire to use X.509 Client Certificates:

1 Configure Spotfire to use HTTPS; for instructions, see “Configuring HTTPS” on page 95.

2 Configure Spotfire to use X.509 Client Certificate Authentication; for instructions, see “Configuring X.509 Client Certificate Authentication” on page 97.

4.8 Two-Factor Authentication

Spotfire Server supports one form of two-factor authentication. It is possible to combine the chosen primary authentication method with X.509 Client Certificates. Typically, the primary authentication method in the two-factor authentication is BASIC, but it is also possible to use the other authentication methods.

When two-factor authentication is enabled, the server requires the name of the authenticated user to match the user name in the provided X.509 certificate.

Note: It is not possible to use two-factor authentication if the Spotfire Web Player is set up to use Kerberos with Delegation.

To enable two-factor authentication, first configure the server to use the chosen primary authentication method.

Then select **Enable two-factor authentication** on the **Authentication** panel in the **Configuration** tab of the Configuration Tool. This creates a second authentication panel.

Then configure the server to use client certificates in the second authentication panel.

OR

Use the command line tool to set up the primary authentication method and the client certificates and then use the following config-two-factor-auth command:

```
config-two-factor-auth --enabled=true
```

4.9 External Authentication Method

Spotfire clients may access Spotfire Server through an external authentication mechanism, a proxy or a load balancer.

When using an external authentication mechanism, Spotfire Server gets the external user name from an HTTP header or a cookie. To get the external user name from an HTTP header or a cookie could potentially be a security risk and it is strongly recommended to restrict the permissions to use this feature. It is also recommended only to use the External Authentication Method when using a load balancer or proxy.

When configuring the External Authentication Method, you can add several constraints:

- It is possible to configure Spotfire Server to allow the External Authentication Method only when using a secure (SSL) connection.
- It is possible to specify allowed hostnames and/or IP addresses of the client computers that are permitted to log in via the External Authentication Method. You can list allowed IP's and/or write regular expressions, if you specify both, the TSS will first check in the list and then the regular expression.

In some cases, the proxy or load balancer has already forced the client to authenticate itself. Some proxies or load balancers are capable of forwarding the name of the authenticated user to Spotfire Server. By enabling the External authentication method on Spotfire Server, it can extract the identity of the client so that the client doesn't have to authenticate twice. Any proxy or load-balancer that can propagate the user name, so that it is available in the HTTP request to the server as a request attribute, is compatible.

Typical scenarios are:

- When both Spotfire Server cluster and its load balancer are configured for NTLM authentication.
- When a load balancer is configured for X.509 Client Certificate authentication and propagates the user names extracted from the certificates.

The External authentication method may be used as a supplementary authentication method that can be used together with the main authentication method, but it can also be used as the main and only authentication method.

- If the external authentication method is to be used as the only authentication method, this shall also be specified in the Authentication Panel.

- If clients are always supposed to go through a load balancer to reach Spotfire Server, configure external as the main authentication method. In this case it is not possible to access a Spotfire Server directly.
- Even if a load balancer is used in front of a set of Spotfire Server instances, accessing the server directly may be desired. If this is the case, configure another authentication mechanism (any mechanism is allowed) as the main authentication method and external as a supplementary authentication method.

Use the Configuration Tool or the **config-external-auth** command (page 207) to set up and enable the External authentication method:

Note: In Spotfire Server 6.0, the **config-delegate-auth** command was replaced by the config-external-auth command. Old scripts using config-delegate-auth will still work.

Enable External Authentication (required)	Specifies whether or not the External authentication method should be enabled.
Source	**Attribute**: Enter the name of the HTTP request attribute that contains the name of the authenticated user. **Header**: Enter the name of the HTTP request header that contains the name of the authenticated user **Cookie**: Enter the name of the HTTP request cookie that contains the name of the authenticated user. **Authentication Filter**: Retrieves the user name from the getUserPrincipal() method of javax.servlet.http.HttpServletRequest.
Require SSL	Select yes for external authentication only to be available for SSL connections.
Allowed host (hostname or IP address)	A list of hostnames and/or IP addresses of the client computers that are allowed to perform external authentication. If no allowed hosts are specified, all client computers are permitted to perform external authentication.
Allowed IP:s (regular expression)	Add a regular expression that should match the IP addresses of remote hosts that are permitted to perform External authentication. The regular expression shall be written in the syntax supported by java.util.regex.Pattern.

Name filter expression (optional)

A regular expression that can be used to filter the user name extracted from the specified request attribute. The value of the regular expression's first capturing group will be used as the new user name.

Note: In previous releases this option was typically used for extracting the user name from a composite name containing both user name and domain name. Since Spotfire Server now requires the domain name as part of the user name, old configured filter expressions must be updated.

Lower case conversion (optional)

Specifies whether or not to convert the propagated user name to lower case. The default is not to convert to lower case.

4.10 Impersonation

What Is Impersonation?

When Spotfire Servers are used in conjunction with one or more Spotfire Web Player servers, which have been configured for certain authentication methods, for instance NTLM, impersonation also needs to be enabled on Spotfire Servers for seamless login.

Impersonation means that the Spotfire Web Player is responsible for authenticating users. Calls from the Spotfire Web Player to Spotfire Server cluster will be made on behalf of the person authenticated. For example, consider the case when the Spotfire Web Player server is configured for certificate authentication. This authentication method is done on the HTTPS network level and there is no password or token which can be conveyed to Spotfire Server cluster for login. Instead the Spotfire Web Player server is trusted for impersonation. The Spotfire Web Player server is allowed to make calls on behalf of any user without the ordinary authentication mechanism. This means the user will see his/her specific files in the library etc.

Enabling impersonation can be a potential security issue, which is why this is disabled by default. To strengthen security there are a number of requirements that can be imposed on a call in order for it to be allowed to impersonate.

Enabling Impersonation

The call from a Spotfire Web Player server to Spotfire Server cluster will always require authentication. This is done as a certain user that has been specified in the configuration of the Spotfire Web Player server. Users that should be able to impersonate must be members of the Impersonator group. It is recommended that these users do not have additional privileges.

The Impersonator group can have many users, add the same user as configured on the Spotfire Web Player server. See the *TIBCO Spotfire Web Player: Installation and Configuration Manual* for more information.

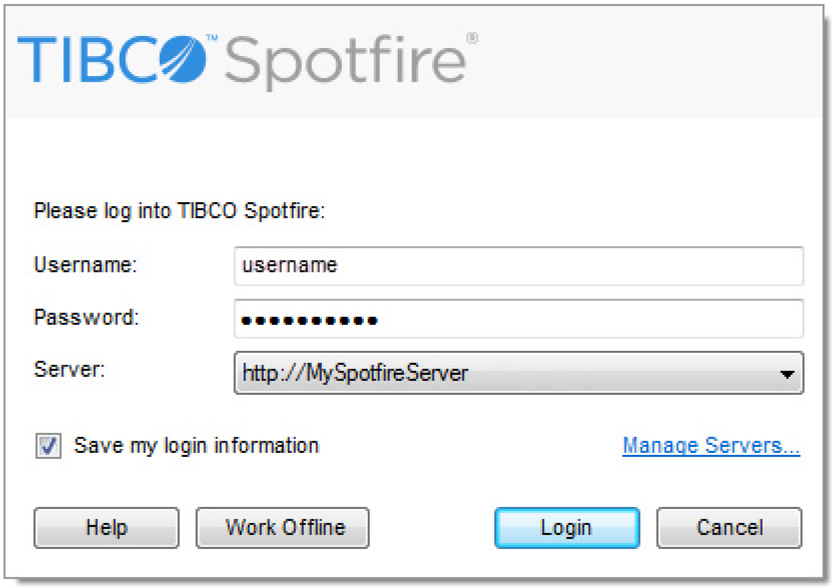
Specific requirements can also be made on the origin of an impersonate call. Typically, you would want to configure Spotfire Server cluster to only allow impersonation calls originating from the machines running a trusted Spotfire Web Player server.

If one or more servers are listed in the **Allowed hosts** fields, only calls originating from these machines are allowed. Allowed machines can be specified in two ways: originating IP number or originating name. The originating IP number should be the IP number of the machine, and a specified originating name is resolved to one (or more) IP numbers using DNS. Only calls originating from one of the mentioned machines are valid for impersonation. If no information is provided in the Web Player Server field, then calls originating from any machine are valid for impersonation.

You can also require HTTPS. All the requirements you decide to set up must be met for the impersonation call to be allowed.

4.11 Configuring Login Behavior

Configure how login shall work; here is a picture of the Spotfire client login dialog.



Please log into TIBCO Spotfire:

Username: username

Password: ●●●●●●●●●●●

Server: http://MySpotfireServer

☑ Save my login information

You can configure the following options for the Spotfire client login dialog:

- If the login dialog should be displayed.
- If users should be allowed to work offline or if they always must log in.
- If users can select “Save my login information” in the login dialog and store the login information for future automatic log in.
- If users should be forced to log in after working offline for a certain number of days.
- If you want an RSS feed to be shown in the login dialog.
- If users should be able to enter their own credentials in the login dialog.

Configuring RSS Feed

Spotfire Server can be configured to display messages to the end users in the login dialog, like news of upcoming scheduled maintenance.

One option is to specify a path to an **rss.xml** file located on an Spotfire Server, which can be updated manually. Another is to specify the URL to an external RSS feed. You must make sure the specified RSS feed complies with the standard RSS 2.0 specification, and that the source is available to the end users' clients. HTML in the RSS feed is not supported.

To enable all users see the news in the login dialog, set the Display behavior setting to **Always**. The login dialog will be shown to all users regardless of whether they have opted to save their login credentials for automatic login.

▸ To configure the login behavior:

Use the command **config-login-dialog** (page 220) and restart the server.
Note: Only use **--show-login-dialog** with the **never** option together with single sign-on methods: NTLM, Kerberos, and X.509 Client Certificates.

4.12 External Directories and Domains

Spotfire Server stores information about the user accounts in the Spotfire database. It can integrate with either external directories like LDAP directories or Windows domains, where the existing environment can be integrated. Meaning that Spotfire Server does not need to manage all information about the users and their credentials by itself. Users and groups from LDAP directories or Windows domains are called external users and groups.

Starting with version 5.0, Spotfire Server keeps track of which domain every user belongs to. A user created by an administrator directly within Spotfire Server using the Administration Console or the Administration Manager in the Spotfire client, belongs to the SPOTFIRE domain. The SPOTFIRE domain is also known as the internal domain and is used by the User Directory when it is running in database mode. An external user belongs to a domain with the same name as the domain it belongs to in the external directory. All Spotfire User Interfaces that display user names will also display the domain names as part of the user names, except for users belonging to the internal SPOTFIRE domain.

The supported external directories can have domain names in two forms:

- DNS domain names (for example **research.example.com**, a complete user name will be of the form **someone@research.example.com**) and
- NetBIOS domain names (for example **RESEARCH**, a complete user name will be of the form **RESEARCH\someone**)

When configuring Spotfire Server, the desired **domain name style** must be set before the server is started for the first time. Which domain name style to select is highly dependent on the combination of authentication method and User Directory mode that is intended to set up.

Note: Be careful when selecting domain name style for your system; it will affect what information Spotfire Server will store within the Spotfire database. The domain name style can be changed using the switch domain name style command if the User Directory is in LDAP mode and is synchronizing with an Active Directory Server. For other User Directory modes, there are no tools to alter that information if the domain name style later needs to be changed.

Below is a matrix showing which domain name style to use for different combinations of authentication method and User Directory mode. Combinations not supported are marked " - ".

Spotfire Server will warn and even refuse to start if you try to set up an authentication method and a User Directory with incompatible domain name styles. If you for some reason need to go ahead with an officially incompatible configuration, you will need to set the **allow incompatible domain name styles** configuration property to make the server start at all. One such reason could be a custom Post-Authentication Filter which creates a bridge between the two originally incompatible domain name styles. (The **allow incompatible domain name styles** option can be set using the **config-userdir** command).

Collapse Domains Configuration Property Enabled

		User Directory mode			
		Database	**LDAP/AD**	**LDAP/other**	**Windows NT**
Authentication Method	**Basic Database**	NetBIOS(DNS)	-	-	-
	Basic/LDAP/AD	NetBIOS(DNS)	NetBIOS(DNS)	NetBIOS(DNS)	-
	Basic/LDAP/other	NetBIOS(DNS)	NetBIOS(DNS)	NetBIOS(DNS)	-
	Basic/Windows NT	-	-	-	NetBIOS(DNS)
	NTLM	NetBIOS(DNS)	NetBIOS(DNS)	NetBIOS(DNS)	-
	Kerberos	NetBIOS(DNS)	NetBIOS(DNS)	NetBIOS(DNS)	-
	X.509 Client Certs	NetBIOS(DNS)	NetBIOS(DNS)	NetBIOS(DNS)	-

- Unsupported combination of authentication method and User Directory mode.

Note: NetBIOS is the recommended domain name style, but DNS will also work.

Collapse Domains Configuration Property Not Enabled

		User Directory mode			
		Database	**LDAP/AD**	**LDAP/other**	**Windows NT**
Authentication Method	**Basic Database**	NetBIOS, DNS	-	-	-
	Basic/LDAP/AD	NetBIOS, DNS	NetBIOS, DNS	#	-
	Basic/LDAP/other	NetBIOS, DNS	#	DNS	-
	Basic/Windows NT	-	-	-	NetBIOS
	NTLM	NetBIOS, DNS	NetBIOS, DNS	#	-
	Kerberos	NetBIOS, DNS	NetBIOS, DNS	DNS	-
	X.509 Client Certs	NetBIOS, DNS	NetBIOS, DNS	DNS	-

-	Unsupported combination of authentication method and User Directory mode.
#	For this combination of authentication method and User Directory mode it is recommended to enable the collapse domains option.

A consequence of the new domain tracking is that the users may have to provide the domain names as part of their user names when logging in to Spotfire Server. For the Basic/LDAP and Basic/Windows NT authentication methods, the setting of the **wildcard domain** configuration property decides how the server maps a user to a domain during authentication. When the **wildcard domain** configuration property is enabled (it is by default), Spotfire Server will check if the user name contains a domain name, and if it does, that domain name will be used. If not, the server will attempt to authenticate the user with the provided user name and password in every domain it knows about, until the combination of domain name, user name and password results in a successful authentication attempt, or until there are no more domain names to try. If the **wildcard domain** configuration property is turned off, the domain name must be specified by the user unless it belongs to the configured **default domain**. This can be configured in the Configuration Tool.

Note: If the **wildcard domain** configuration property is enabled and two identically named users in different domains have the same password, there is a risk that the wrong account is selected when one of these users log in. Thus, if security has a higher priority than user convenience, make sure to turn the **wildcard domain** configuration property off. There is also a risk that multiple authentication attempts will lock out the “correct” user.

Spotfire Server provides a configuration property that reverts to the behavior from previous releases. The configuration property is called **collapse domains** and enabling this means that the external domain of a user will be essentially ignored and that different users with the same user name, but in different domains, will share account on Spotfire Server. When the **collapse domains** configuration property is enabled, all external users and groups will be associated with the SPOTFIRE domain, regardless of which domain they belong to in the external directory. If you want to keep running Spotfire Server without ever caring about domain names, enable both the **collapse domains** and **wildcard domain** configuration properties. Doing so will ensure that all users will belong to the internal SPOTFIRE domain and no users will have to enter a domain name when logging in. (The **collapse domains** configuration property can be set in the Configuration Tool or by using the **config-userdir** command).

Note: All users will belong to one domain when the **collapse domains** configuration property is enabled. If there are multiple users with the same account name in different external domains, they will now effectively share the same account within Spotfire Server. If security has a higher priority than user convenience, make sure not to enable the **collapse domain** configuration property.

Note: It is not recommended to change the **collapse domains** configuration property after once having synchronized Spotfire Server with an external directory. Doing so will lead to double accounts with different domain names for every synchronized user and group in the User Directory. The new accounts will not inherit the permissions of the old accounts.

4.13 User Directory Modes

The User Directory mode is where the User Directory retrieves information about users and groups. The supported User Directory modes are:

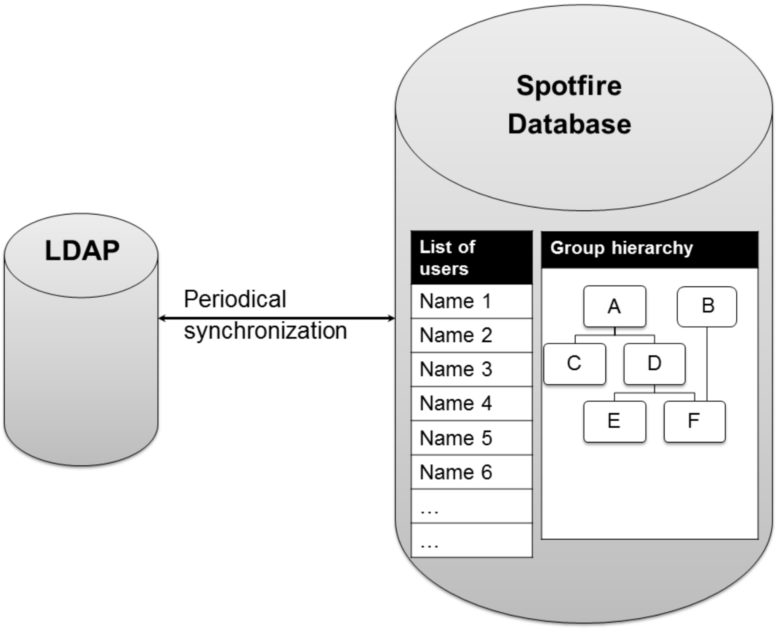
- Spotfire database Mode
- LDAP Mode
- Windows NT Mode

4.13.1 User Directory in Spotfire Database Mode

This User Directory mode only uses the user names stored in the Spotfire database and does not use any external directory for user and group information. Users are added manually to Spotfire Server via the Administration Console or through automatic registration via the Post-Authentication Filter in auto-creating mode. It is also possible to import users and groups, see commands “import-groups” on page 250 and “import-users” on page 253.

This is the default User Directory mode, which does not require any additional configuration. This configuration is easy and fast to set up and is recommended for small sites.

4.13.2 User Directory in LDAP Mode



Spotfire Server supports the following LDAP servers:

- Microsoft Active Directory
- Oracle Directory Server, Sun Java System Directory Server, Sun ONE Directory Server, iPlanet Directory Server, Netscape Directory Server

When the User Directory is in LDAP mode, the User Directory is synchronized with one or more LDAP directories. In previous releases, the User Directory synchronized LDAP groups in the background but all user lookups in the User Directory resulted in LDAP queries. Starting with Spotfire Server version 5.0, the User Directory will synchronize information about both users and groups in the background. All user and group lookups will now only query the Spotfire database. When upgrading to 5.0 or later, all old schedules for group synchronization will now also be used for user synchronization.

Schedule LDAP Synchronizations

There are two algorithms that can be used when configuring the recurrence of synchronization tasks, one is based on cron schedules and the other on sleep time between synchronizations.

Sleep time is only used when no cron schedule exists for the LDAP configuration. The sleeping period is configurable and by default set to 60 minutes.

New configurations have two default cron schedules: restart and daily. Restart runs synchronization at each restart of Spotfire Server and daily runs synchronization once a day (at midnight server time). Upgraded configurations may not have these default cron schedules.

Each LDAP configuration has its own schedules. It is possible to use cron schedules for one LDAP configuration and sleep time for another.

User Synchronization

By default, the User Directory will only synchronize users (and not groups) from the LDAP directories.

Once an LDAP user has been synchronized and imported to the User Directory, it will permanently be a part of the User Directory. If the LDAP user is later removed from the LDAP directory, the corresponding user in the User Directory will be disabled. Disabled accounts will still be visible within the Spotfire applications but it will not be possible for the user to log in. To prevent users from being disabled by failed synchronization attempts, for example caused by network errors, the **safe synchronization** option can be enabled. When this option is enabled, no users will be disabled just because they could not be found during synchronization. By default, this option is not enabled since it could be a security problem if removed LDAP users are not disabled in the User Directory.

Note: It is usually not possible to log in as a removed LDAP user anyway, as the LDAP directory will block the authentication attempt if it is also responsible for authenticating users.

Users may also be explicitly disabled in the LDAP directories. Such users will always be disabled in the User Directory regardless of whether or not the **safe synchronization** option is enabled.

Group Synchronization

When setting the **enable group synchronization** option the User Directory will synchronize groups from the LDAP directory. This capability is useful to mirror the group hierarchies in the LDAP directory in the User Directory. Synchronizing groups relieves the administrator of the responsibility of managing the group memberships. Assigning licenses and privileges to Spotfire groups is still accomplished in the normal fashion.

Synchronized LDAP groups can not be manually modified in the User Directory. Synchronized groups can be placed into manually created groups in the User Directory and thereby be granted permissions. If an LDAP group has been synchronized and it is removed from the list of groups to synchronize, it will keep the members from the last synchronization, but will revert into a normal group which is possible to modify.

Note: The User Directory does not support cyclic group memberships, where the ancestor of a group is also a descendant of the same group. If the User Directory detects a group membership cycle, it will be broken up arbitrarily.

When configuring he groups to be synchronized, specify either the group account names or the distinguished names. The account names and the distinguished names may contain an asterisk (*) as a wildcard character. This wildcard behaves just like the asterisk wildcard in standard LDAP search filters. It is also possible to specify the distinguished name of an LDAP container containing one or more groups. All those groups will then be synchronized. It is possible to mix all variants.

Note: If the enable group synchronization configuration property is set and no groups or group context names are configured, the User Directory will synchronize all groups it can find in the configured context names.

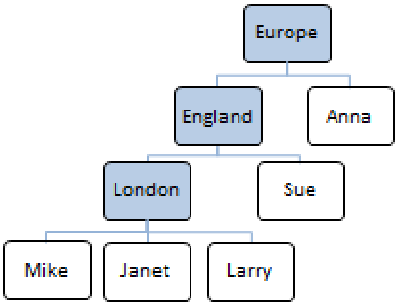
Filtering Users By Groups

The synchronized groups can also be used to filter the set of users that are to be synchronized with the User Directory. By enabling the **filter users by groups** option, only users that are members in at least one of the synchronized groups will be synchronized with the User Directory.

Group-Based and Role-Based Synchronization

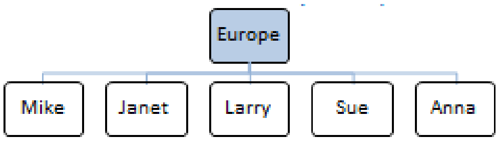
- For Active Directory servers Spotfire Server can synchronize groups.
- For the Directory Server product family (Oracle Directory Server, Sun Java System Directory Server, Sun ONE Directory Server, iPlanet Directory Server and Netscape Directory Server) Spotfire Server can synchronize either groups or roles.

Here are examples of default behavior of **group-based** and **role-based** group synchronization. The examples are based on the figure below.

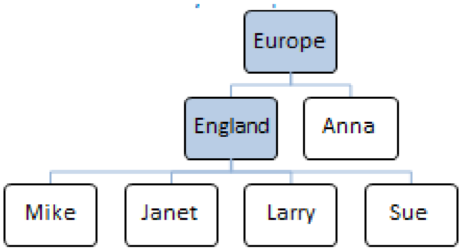


Group-Based Synchronization

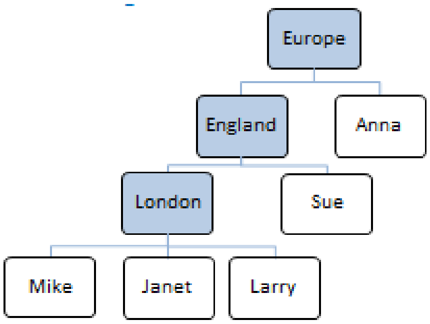
1 If you only specify the group "Europe" to be synchronized in your LDAP configuration, the User Directory will synchronize according to the figure below. The groups England and London will not be visible, but will automatically be replaced with their members:



2 If you specify the groups "Europe" and "England" to be synchronized in your LDAP configuration, the User Directory will synchronize according to the figure below. The group London will not be visible, but will automatically be replaced with its members:

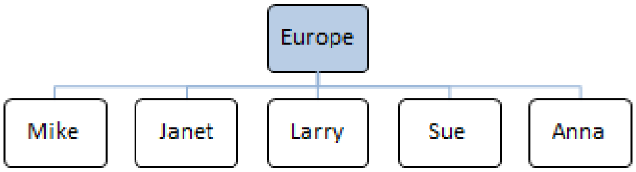


3 If you specify the groups "Europe", "England" and "London" explicitly to be synchronized in your LDAP configuration, the User Directory will synchronize according to the figure below:



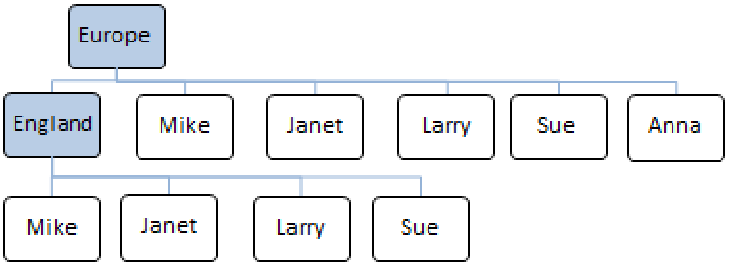
Role-Based Synchronization

1 If you only specify the role "Europe" to be synchronized in your LDAP configuration, the User Directory will synchronize according to the figure below. The roles England and London will not be visible, but will automatically be replaced with their members:

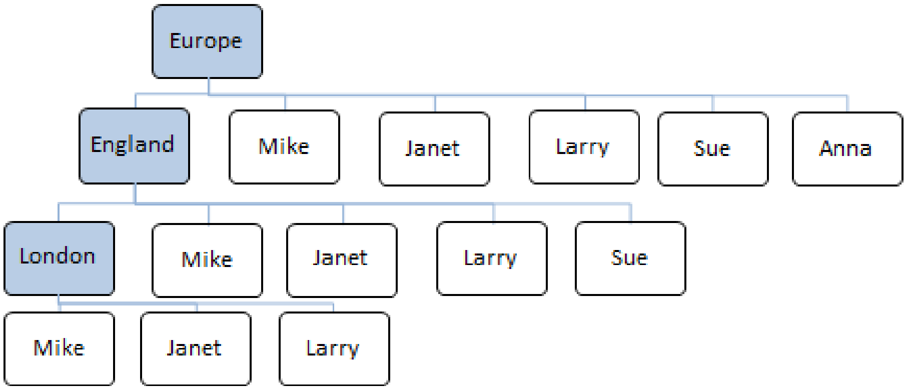


2 If you specify the roles "Europe" and "England" to be synchronized in your LDAP configuration, the User Directory will synchronize according to the figure below. The role London will not be visible. Due to the nature of roles in the Directory Server

product family, every role will automatically include all direct members as well as all members of sub roles:



3 If you specify the roles "Europe", "England" and "London" explicitly to be synchronized in your LDAP configuration, the User Directory will synchronize according to the figure below. Due to the nature of roles in the Directory Server product family, every role will automatically include all direct members as well as all members of sub roles:



The member and the memberOf Algorithms

There are two algorithms to choose from when configuring group synchronization; the member and the memberOf algorithms. The memberOf algorithm relies on a calculated attribute in the LDAP directory and may induce more load on the LDAP servers. Not all LDAP directories have support for the memberOf algorithm. The member algorithm performs significantly more LDAP queries, but with much smaller result sets than the memberOf algorithm. See the recommendations below for group synchronization set ups use depending on different LDAP servers.

Recommendations

Using a Microsoft Active Directory server:

- Configure group-based synchronization with the memberOf algorithm.

Using a Sun Java System Directory Server (version 6 and later):

- Configure group-based synchronization with the memberOf algorithm or
- Configure role-based synchronization with the memberOf algorithm

Using a Sun ONE Directory Server (version 5 and earlier):

- Configure role-based synchronization with the memberOf algorithm or
- Configure group-based synchronization with the member algorithm

Note: Configuring group-based synchronization with the memberOf algorithm will not work on Sun ONE Directory Servers, nor will role-based synchronization work with the member algorithm.

4.13.2.1 LDAP Directory Commands

The commands used to set up authentication mode and User Directory mode with an LDAP directory are listed below.

Commands for creating and updating an LDAP configuration:

config-ldap-group-sync	Configures group synchronization for an LDAP configuration.
create-ldap-config	Creates a new LDAP configuration to be used for authentication and/or the User Directory LDAP provider.
update-ldap-config	Updates LDAP configurations.

Commands referencing existing LDAP configurations:

config-ldap-userdir	Configures the LDAP User Directory mode.
list-ldap-config	Displays LDAP configurations.
remove-ldap-config	Removes LDAP configurations.
set-auth-mode	Sets the authentication mode
set-userdir-mode	Sets the User Directory mode

4.13.2.2 LDAP Authentication and User Directory Settings

The following information is required to set up LDAP authentication and User Directory mode, including LDAP group synchronization. Contact the LDAP directory administrator if you do not have the required information.

The following table provides an overview of LDAP settings and their applicability. Detailed descriptions of the settings are provided below the table.

- **A**: Applicable to LDAP as authentication mechanism
- **UD**: Applicable to LDAP User Directory mode
- **GS**: Applicable to LDAP User Directory mode with group synchronization
- **M**: Mandatory

- **: Required by configurations with LDAP server type **Custom**. These options have template values for the non pre-defined LDAP server types. The template values can be overridden when necessary.

A			**Authentication Attribute** Specifies the name of the LDAP attribute containing a user identity that can be used for authenticating with the LDAP server.
A	UD	M	**LDAP Server Type** Specifies the type of LDAP server: **ActiveDirectory**, **SunOne**, **SunJavaSystem** or **Custom**.
A	UD	M	**LDAP Server URLs** A white-space separated list of LDAP server URLs.
A	UD	M	**Context Names** A list of distinguished names (DNs) of the containers holding the user accounts to be visible within Spotfire Server.
A	UD		**Username** The name of the LDAP service account to be used when searching for users and groups in the LDAP directory.
A	UD		**Password** The password for the LDAP service account.
A	UD		**Security Authentication** Specifies the security level to use when binding to the LDAP server. The default value is **simple**.
A	UD	**	**User Search Filter** Specifies an LDAP search expression filter to be used when searching for users.
A	UD		**Referral Mode** Specifies how LDAP referrals should be handled.
A	UD	**	**Username Attribute** Specifies the name of the LDAP attribute containing the user account names.
A	UD		**Custom LDAP Properties** Multiple key-value pairs specifying additional JNDI environment properties to be used when connecting to the LDAP server.
	UD		**Request Control** Specifies the type of LDAP controls to be used when executing search queries to the LDAP server: **Probe**, **PagedResultsControl**, **VirtualListViewControl** or **none**.

	UD		**Page Size** Specifies the page size to be used with the paged results control or the virtual list view control when performing search queries to the LDAP server. The page size value defaults to **1000** for both the paged results control and the virtual list view control.
	UD		**Import Limit** Specifies a threshold that limits the number of users that can be imported from an LDAP server to Spotfire Server in one query.
	UD		**Synchronization Schedules** Specifies a list of schedules for when the synchronization task should be performed.
	GS		**Group Synchronization Enabled** Specifies whether or not group synchronization should be enabled for this LDAP configuration.
	GS		**Group Names** Specifies a list of distinguished names (DNs) of either individual groups to be synchronized or a context name where all groups are to be synchronized. If the **group synchronization enabled** option is set and the list of group names is empty, then all groups that can be found in the LDAP directory will be synchronized.
	GS	**	**Group Search Filter** Specifies an LDAP search expression filter to be used when searching for groups.
	GS	**	**Group Name Attribute** Specifies the name of the LDAP attribute containing the group account names.
	GS	**	**Supports memberOf** Specifies whether or not the LDAP servers support a **memberOf**-like attribute on the user accounts that contain the names of the groups or roles that the users are members of. In general, this is true for all Microsoft Active Directory servers and all types of Sun directory servers.
	GS	**	**Member Attribute** For all LDAP servers with support for a **memberOf**-like attribute, this option specifies the name of the LDAP attribute on the user account that contains the names of the groups or roles that the user is a member of.
	GS	**	**Ignore Member Groups** Specifies whether or not the group synchronization mechanism should recursively traverse the synchronized groups' non-synchronized subgroups and include their members in the search result.

Authentication Attribute

Specifies the name of the LDAP attribute containing a user identity that can be used for authenticating with the LDAP server. This attribute fills no purpose in most common LDAP configurations, but can be useful in more advanced setups, where the distinguished name (DN) does not work for authentication or where users should be able to login using a user name which does not map directly to an actual LDAP account.

A typical case for using this option is when setting up SASL, see “Configuring SASL Authentication for LDAP” on page 58.

LDAP Server Type

Specifies the type of LDAP server. There are four valid types: **ActiveDirectory**, **SunOne**, **SunJavaSystem**, and **Custom**.

When specifying one of the pre-defined server types, we will assume that default values will be applied for the most fundamental configuration options. It is possible to override the default values. When specifying a **Custom** LDAP server type, there is no configuration template and all fundamental configuration options must be specified explicitly. The table above shows which configuration options are required for a **Custom** LDAP server type.

LDAP Server URLs

A whitespace-separated list of LDAP server URLs. An LDAP server URL has the format **<protocol>://<server>[:<port>]**:

- **<protocol>**: Either LDAP or LDAPS
- **<server>**: The fully qualified DNS name of the LDAP server.
- **<port>**: An optional number indicating the TCP port the LDAP service is listening on. When using the LDAP protocol, the port number defaults to **389**. When using the LDAPS protocol, the port number defaults to **636**. Active Directory LDAP servers also provides a Global Catalog containing forest-wide information, instead of domain-wide information only. The Global Catalog LDAP service by default listens on port number **3268** (LDAP) or **3269** (LDAPS).

Spotfire Server does not expect any search base, scope, filter or other additional parameters after the port number in the LDAP server URLs. Such properties are specified using other configuration options for this command.

Examples: LDAP server URLs

```
LDAP://myserver.example.com

LDAPS://myserver.example.com

LDAP://myserver.example.com:389

LDAPS://myserver.example.com:636

LDAP://myserver.example.com:3268

LDAPS://myserver.example.com:3269
```

Context Names

A list of distinguished names (DNs) of the containers holding the LDAP accounts to be visible within Spotfire Server. When specifying more than one DN, the DNs must be separated by pipe characters (|). If the specified containers contain a large number of users, but only a few should be visible in Spotfire Server, a custom user search filter can be specified to include only the filtered users, See “User Search Filter” on page 90.

Username

The name of the LDAP service account to be used when searching for users and groups in the LDAP directory. This service account does not need to have any write permissions, but it needs to have read permissions for all configured context names (LDAP containers). For most LDAP servers, the account name is the account’s distinguished name (DN). For Active Directory, the account name can also be specified in the forms **ntdomain\name** or **name@dnsdomain**.

Examples:

CN=spotsvc,OU=services,DC=research,DC=example,dc=COM

RESEARCH\spotsvc (: Active Directory only)

spotsvc@research.example.com (: Active Directory only)

Password

The password for the LDAP service account.

Security Authentication

Specifies the security level to use when binding to the LDAP server. The default value is **simple**. Only use this parameter in special cases, and use it with care in production environments.

- To enable anonymous binding, it should be set to **none**.
- To enable plain user name/password authentication, it should be set to **simple**.
- To enable SASL authentication, it should be set to the name of the SASL mechanism to be used. Spotfire Server supports the two SASL mechanisms **DIGEST-MD5** and **GSSAPI**. You can set multiple **-C** flags to set the additional JNDI environment properties that the SASL authentication mechanism typically requires.

A typical case for using this option is when setting up SASL, see “Configuring SASL Authentication for LDAP” on page 58

User Search Filter

This parameter specifies an LDAP search expression filter to be used when searching for users.

If only a subset of all the users in the specified LDAP containers should be allowed access to Spotfire Server, a restrictive user search filter can be specified. For instance, the search expression can be configured so that it puts restrictions on which groups the users belong to, or which roles they have.

- For Active Directory servers, the parameter value defaults to **objectClass=user**.
- For Active Directory servers, access can be restricted to only those users belonging to a certain group by using a search expression with the pattern

&(objectClass=user)(memberOf=<groupDN>) where **<groupDN>** is to be replaced by the real DN of the group to which the users must belong. If the users are divided among multiple groups, use the pattern **&(objectClass=user)(|(memberOf=<firstDN>)(memberOf=<secondDN>))**. Add extra **(memberOf=<groupDN>)** sub-expressions as needed.

Example: **&(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)**

- For any version of the Sun Directory Servers, it defaults to **objectClass=person**.
- For a Sun Java System Directory Server version 6 and later, the same effect can be achieved by using a search expression with the pattern **&(objectClass=person)(isMemberOf=<groupDN>)**. If the users are divided among multiple groups, use the pattern **&(objectClass=person)(|(isMemberOf=<firstDN>)(isMemberOf=<secondDN>))**. Add extra **(isMemberOf=<groupDN>)** sub-expressions as needed.

  Example: **&(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)**

- For the Directory Server product family, access can be restricted to only those users having certain specific roles. The search expression for role filtering must match the pattern **&(objectClass=person)(nsRole=<roleDN>)**. If multiple roles are of interest, use the pattern **&(objectClass=person)(|(nsRole=<firstDN>)(nsRole=<secondDN>)**. Add extra **(nsRole=<roleDN>)** sub-expressions as needed.

  Example: **&(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)**

The syntax of LDAP search expression filters is specified by RFC 4515. Please consult this specification for information about more advanced filters.

Referral Mode

This argument specifies how LDAP referrals should be handled. Valid arguments are **follow** (automatically follow any referrals), **ignore** (ignore referrals) and **throw** (fail with an error). The default and recommended value is **follow**.

Username Attribute

Specifies the name of the LDAP attribute containing the user account names. For Active Directory servers the value defaults to **sAMAccountName**. For the Directory Server product family with a default configuration, it defaults to **uid**.

Custom LDAP Properties

Multiple key-value pairs specifying additional JNDI environment properties to be used when connecting to the LDAP server. For instance, specifying the key **java.naming.security.authentication** and the value **simple** have the same result as setting the **Security Authentication** option to **simple**.

Request Control

This option determines the type of LDAP controls to be used when executing search queries to the LDAP server. Valid controls are **Probe**, **PagedResultsControl**, **VirtualListViewControl** and **none**.

The default behavior is to probe the LDAP server for the best supported request control. The paged results control is always preferred, since it provides the most efficient way of retrieving the result of the query. The virtual list view control can also be used to retrieve a large number of users, if the paged results control is not supported. The virtual list view control will automatically be used together with a sort control. Both the paged results control and the virtual list view control support a configurable page size, as specified by the **page size** option.

Page Size

This argument specifies the page size to be used with the paged results control or the virtual list view control when performing search queries to the LDAP server. The page size value defaults to **1000** for both the paged results control and the virtual list view control.

Import Limit

This argument specifies a threshold that limits the number of users that can be imported from an LDAP server to Spotfire Server in one query. This can be used to prevent accidental flooding of Spotfire Server's User Directory when integrating with an LDAP server with tens or even hundreds of thousands of users. By setting an import limit, the administrator can be sure that an unexpected high number of users won't affect the server's performance. By default, there is no import limit. To explicitly request unlimited import, set the parameter value to **-1**. All positive numbers are treated as an import limit. Leave this parameter untouched. in most cases.

Group Synchronization Enabled

Specifies whether or not group synchronization should be enabled for this LDAP configuration.

Group Names

Specifies the groups to be synchronized. Groups can be specified with either their account names or their distinguished names (DNs). The account names and the distinguished names may contain an asterisk (*) as a wildcard character. This wildcard behaves just like the asterisk wildcard in standard LDAP search filters. Wildcards work for both account names and distinguished names.

It is also possible to specify the distinguished name of an LDAP container containing multiple groups and thereby synchronizing all those groups. Wildcards can also be used for specifying group containers.

It is possible to mix all variants above.

Consider the following when specifying a group to be synchronized:

- Specify either the group´s account name or its distinguished name (DN). The account name must match the value of the configured **group name attribute**.
- It is possible to use an asterisk (*) as a wildcard character s in the account nameswhen specifying group names. If a configured group name contains wildcard characters and matches multiple groups in the directory, all those groups will be synchronized.

- It is also possible to specify the distinguished name of an LDAP container containing one or more groups. All those groups will then be synchronized.
- It is possible to mix all variants.

Note: If the **enable group synchronization** configuration property is set and the list of group names is empty, then all groups that can be found in the configured context names in the LDAP directory will be synchronized.

Synchronization Schedules

Specifies a list of schedules for when the group synchronization task should be performed. The schedules are specified in the **cron** format, where each schedule consists of either five fields or one shorthand label.

The five fields are, from left to right, with their valid ranges:

- **minute (0-59)**
- **hour(0-23)**
- **day of month (1-31)**
- **month (1-12)**
- **day of week (0-7, where both 0 and 7 indicate Sunday)**

A field may also be configured with the wildcard character (*), indicating that any moment in time matches this field. A group synchronization is triggered when all fields match the current time. If both day of month and day of week have non-wildcard values, then only one of them has to match.

There are also the following shorthand labels that can be used instead of the full **cron** expressions:

@yearly or @annually: run once a year (equivalent to 0 0 1 1 *)

@monthly: run once a month (equivalent to 0 0 1 * *)

@weekly: run once a week (equivalent to 0 0 * * 0)

@daily or @midnight: run once a day (equivalent to 0 0 * * *)

@hourly: run once an hour (equivalent to 0 * * * *)

@minutely: run once a minute (equivalent to * * * * *)

@reboot or @restart: run every time Spotfire Server is started

Refer to the Wikipedia overview article on the cron scheduler.

Group Search Filter

This parameter specifies an LDAP search expression filter to be used when searching for groups.

- For Active Directory servers, the parameter value defaults to **objectClass=group**
- For Oracle Directory Servers and Sun Java System Directory Servers, it defaults to **objectClass=groupOfUniqueNames**

- For Sun ONE Directory Servers, it defaults to **&(|(objectclass=nsManagedRoleDefinition)(objectClass=nsNestedRoleDefinition))(objectclass=ldapSubEntry)**

Group Name Attribute

Specifies the name of the LDAP attribute containing the group account names:

- For Active Directory servers the value defaults to **sAMAccountName**
- For any version of the Sun directory servers with a default configuration, it defaults to **cn**

Supports memberOf

Specifies whether or not the LDAP servers support a **memberOf**-like attribute on the user accounts that contain the names of the groups or roles that the users are members of. In general, this is true for all Microsoft Active Directory servers and the Directory Server product family.

For some LDAP servers with configurations of type Custom, there is no **memberOf**-like attribute. This is declared by setting the supports **memberOf** configuration property to false.

Member Attribute

This parameter value can be set to: **memberOf**, **nsRole** or **isMemberOf**.

For LDAP configurations with the supports memberOf option set to false, the **member attribute** option specifies the name of the LDAP attribute on the group accounts that contains the distinguished names (DNs) of its members. In general, this includes LDAP servers with configurations of type Custom and any Sun ONE Directory Servers (version 5 and earlier) when used with group-based synchronization.

For LDAP configurations with the supports memberOf option set to true, the member attribute option specifies the name of the LDAP attribute on the user accounts that contains the names of the groups or roles that the users are members of. In general, this includes all Microsoft Active Directory server and all types of Sun Directory Servers version 6 and later. For Sun ONE Directory Servers (version 5 and older), this also applies for roles.

- For Microsoft Active Directory servers, the member attribute value defaults to **memberOf**
- For Sun ONE Directory Servers, the member attribute option defaults to **nsRole**
- For Sun Java System Directory Server version 6.0 or later, the member attribute option defaults to **isMemberOf**. To use the roles with the Sun Java System Directory Server or later, it is recommended to use the SunONE configuration template instead.

Note: All configurations with the **memberOf** option set to **false** will use a far less efficient group synchronization algorithm that will generate more traffic to the LDAP servers, because Spotfire Server will first have to search for the distinguished names (DNs) of the group members within the groups, and then perform repeated lookups to translate the member DN to the correct account name.

Ignore Member Groups

This argument determines whether or not the group synchronization mechanism should recursively traverse the synchronized groups' non-synchronized subgroups and include their members in the search result.

For Microsoft Active Directory servers, the parameter value defaults to **false** so that all inherited group memberships are correctly reflected. For any version of the Sun Directory Servers, it defaults to **true**, since the role and groups mechanisms in those servers automatically include those members.

4.13.3 User Directory in Windows NT Mode

Kept for legacy support. Use the Configuration Tool or the command "config-windows-userdir" on page 228 to configure a User Directory in Windows NT Mode.

4.14 HTTPS

By default, Spotfire uses the HTTP protocol for communication between clients and Spotfire Server. To achieve a higher level of security, we recommend using the HTTPS protocol instead, ensuring encryption between clients and server. HTTPS also includes a mechanism for clients to authenticate the server.

To have the server authenticate the clients as well, you can enable X.509 Client Certificate Authentication.

To Enable Encrypted Communication Using HTTPS

- Follow the instructions in section 4.14.1.

To Enable X.509 Client Certificate Authentication

- Start with the instructions in section 4.14.1 and then proceed to the instructions in section 4.14.2.

4.14.1 Configuring HTTPS

Prerequisites

To configure HTTPS, you must first obtain the following:

- A server certificate and private key, stored in a Java keystore (JKS) or PKCS #12 keystore (P12/PFX).

General Procedure

(See the specific instructions following this list.)

1 Stop Spotfire Server (if it is running).

2 Install the keystore containing the server certificate and private key.

3 Configure Spotfire Server to use the HTTPS protocol.

4 Start Spotfire Server.

To stop Spotfire Server:

See “Start and Stop Spotfire Server” on page 110.

To install the keystore containing the server certificate and private key:

- Copy the keystore file to the <**installation dir**>**/tomcat/certs** directory. We suggest using the server's hostname as keystore filename.

To configure Spotfire Server to use the HTTPS protocol:

1 Open the configuration file **<installation dir>/tomcat/conf/server.xml** in an editor and locate the section containing the configuration template for an HTTPS connector:

```
    <!-- Enable this connector if you want to use HTTPS -->
    <!--
    <Connector port="443"
               maxHttpHeaderSize="16384"
               connectionTimeout="30000"
               enableLookups="false"
               URIEncoding="UTF-8"
               disableUploadTimeout="true"
               server="TIBCO Spotfire Server"
               SSLEnabled="true"
               scheme="https"
               secure="true"
               keystoreFile="./certs/[server hostname].jks"
               keystorePass="changeit"
               keystoreType="jks"
               keyAlias="[server hostname]"
               truststoreFile="./certs/[server hostname].jks"
               truststorePass="changeit"
               truststoreType="jks"
               clientAuth="false"/>
-->
```

(In your installation, [**server hostname**] is replaced with the actual hostname of your server.)

Note: The parameters to be changed are shown in bold in the XML snippet above.

2 Remove the lines with the comment markers <!-- and -->.

3 Update the **keystoreFile** parameter with the name of the keystore file containing the server certificate and private key.

4 Set the **keystorePass** parameter to the password for the keystore file containing the server certificate and private key.

5 Set the **keystoreType** parameter to "jks" for a Java keystore or "pkcs12" for a PKCS #12 keystore.

6 Do one of the following:

- If the keystore contains several certificates, the **keyAlias** parameter must be set to the alias for the server certificate and private key.
- If the keystore contains only the server certificate, the line containing the **keyAlias** parameter must be removed.

7 Unless you will enable X.509 Client Certificate Authentication, remove the **truststoreFile**, **truststorePass**, and **truststoreType** parameters.

8 To disable unencrypted HTTP traffic, locate the section containing the default HTTP connector:

```
<Connector port="[HTTP port]"
              maxHttpHeaderSize="16384"
              connectionTimeout="30000"
              enableLookups="false"
              URIEncoding="UTF-8"
              disableUploadTimeout="true"
              server="TIBCO Spotfire Server" />
```

(In your installation, [HTTP port] is replaced with the HTTP port of your server.)

9 Add comment markers <!-- and --> around the HTTP connector configuration:

```
<!--
<Connector port="[HTTP port]"
          maxHttpHeaderSize="16384"
          connectionTimeout="30000"
          enableLookups="false"
          URIEncoding="UTF-8"
          disableUploadTimeout="true"
          server="TIBCO Spotfire Server" />
-->
```

▶ **To start Spotfire Server:**

See “Start and Stop Spotfire Server” on page 110.

4.14.2 Configuring X.509 Client Certificate Authentication

Prerequisites

- Spotfire Server must first be configured for HTTPS as described in the previous section, on page 95.
- A client certificate from a trusted Certification Authority (CA) must be installed on each client. Note that Spotfire Web Player and Spotfire Automation Services are both clients of Spotfire Server.
- You have obtained certificates for the CAs that issued the client certificates.

General Procedure

(See the specific instructions following this list.)

1 Stop Spotfire Server.

2 Install the CA certificates on the server.

3 Configure Spotfire Server to require X.509 Client Certificates for HTTPS.

4 Start Spotfire Server.

To stop Spotfire Server:

See “Start and Stop Spotfire Server” on page 110.

To install the CA certificates:

If you already have a keystore containing the CA certificate(s)

- Copy the keystore file to the <**installation dir>/tomcat/certs** directory.

  Note: The keystore containing the CA certificate(s) can be in either PKCS #12 or JKS format.

If you do not yet have a keystore

1 Create a keystore and import the CA certificate(s). CA certificates can be in either PEM format or DER format.

Execute the following command:

```
> <installation dir>/jdk/bin/keytool -importcert -alias cacert -keystore <installation dir>/
tomcat/certs/<keystore filename> -file <certificate filename>
```

Example: Windows

```
> C:\tibco\tss\6.5.0\jdk\bin\keytool -importcert -alias cacert -keystore C:\tibco\tss\
6.5.0\tomcat\certs\tss.jks -file cacert.cer
```

2 Repeat the previous step for each additional CA certificate.

To configure Spotfire Server to require X.509 Client Certificates for HTTPS:

1 Open the configuration file **<installation dir>/tomcat/conf/server.xml** in an editor.

2 Locate the section containing the configuration for the HTTPS connector:

```
<Connector port="443"
               maxHttpHeaderSize="16384"
               connectionTimeout="30000"
               enableLookups="false"
               URIEncoding="UTF-8"
               disableUploadTimeout="true"
               server="TIBCO Spotfire Server"
               SSLEnabled="true"
               scheme="https"
```

```
secure="true"
keystoreFile="./certs/[server hostname].jks"
keystorePass="changeit"
keystoreType="jks"
keyAlias="[server hostname]"
truststoreFile="./certs/[server hostname].jks"
truststorePass="changeit"
truststoreType="jks"
clientAuth="false"/>
```

Note: The parameters to be changed are shown in bold in the XML snippet above.

3 Update the t**ruststoreFile** parameter with the name of the keystore file containing the CA certificate(s).

4 Set the **truststorePass** parameter to the password for the keystore file containing the CA certificate(s).

5 Set the **truststoreType** parameter to "jks" for a Java keystore or "pkcs12" for a PKCS #12 keystore.

6 Set the **clientAuth** parameter to "true".

▸ **To start Spotfire Server:**

See “Start and Stop Spotfire Server” on page 110.

4.14.3 Using Encryption in a Load-Balanced Environment

In a load-balanced environment, clients communicate with the load balancer using HTTP or HTTPS, which redirects traffic to the servers using the AJP protocol. Because the AJP protocol cannot be encrypted, we recommend that the load balancer and Spotfire Server reside on the same secure network, or that other security measures, such as tunnel technology, be used.

To configure the load balancer to use HTTPS, see “Setting up HTTPS” on page 189.

4.15 Configuring LDAPS

In an LDAP environment, where the Spotfire system communicates with an LDAP directory, it might be a good idea to secure the LDAP protocol using SSL, if the LDAP directory supports this.
To achieve this, you must first set up the LDAP directory server to communicate using SSL. Then you must get Spotfire Server(s) to trust this certificate if you are using a self-signed certificate. This is done by following steps:

1 Export the certificate to file and copy it to Spotfire Server.

2 With a command prompt or shell, navigate to the directory **<installation dir>/jdk/jre/lib/security** and execute the keytool command located in the **<installation dir>/jdk/bin/** directory to import the certificate:

```
../../bin/keytool -import -file ldapserver.crt -keystore cacerts -alias spotfire_ldaps
```

Replace **ldapserver.crt** with the name of the exported certificate.

When prompted, enter the password to the cacerts keystore. The default password is **changeit**.

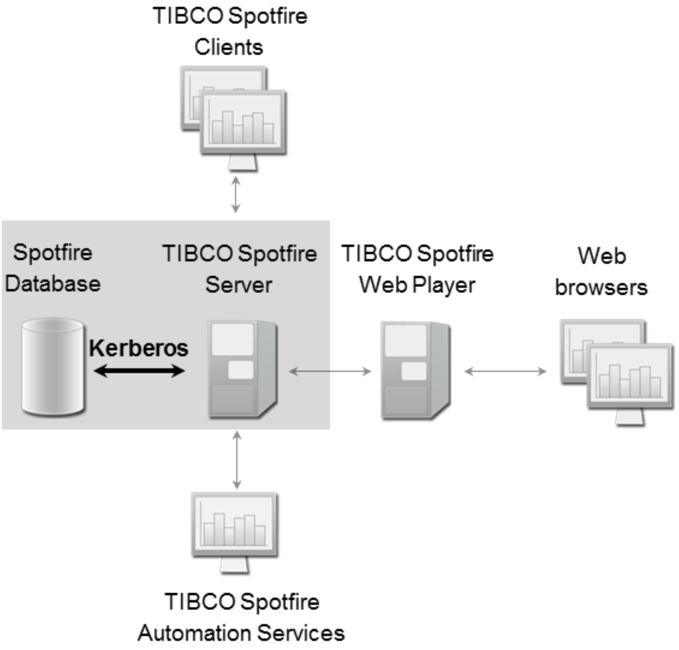
3 Verify that the certificate has been successfully added by using the keytool command:

```
../../bin/keytool -list -keystore cacerts -alias spotfire_ldaps
```

When prompted, enter the password to the cacerts keystore. The result of the command should be that the certificate is added.

Use the **create-ldap-config** (page 231) or the **update-ldap-config** (page 288) command to activate LDAPS.

4.16 Using Kerberos to Log In to the Spotfire Database



To increase security in the Spotfire system, you may want to set up Spotfire Server to authenticate with the Spotfire database using the Kerberos protocol. This only affects how the database connections are authenticated and has nothing to do with clients to Spotfire Server using the Kerberos authentication method.

These instructions assume that the person following them has experience with the database and possibly also with setting up Kerberos in other setups.

Prerequisites

- Windows Domain Controllers running Windows Server 2003 SP1 or later.
- A computer with the **Microsoft Active Directory Users and Computers MMC** snap-in.

- A computer with the **Microsoft Support Tools** installed.
- A domain administrator account or a user account which is a member of the built in **Account Operators** domain group, or any account with equivalent permissions.
- The database server must already be installed and configured for both Kerberos authentication and user name/password authentication.
- Microsoft Active Directory is used as Kerberos environment.
- If the database is an Oracle database, then download Oracle's latest JDBC driver (**ojdbc6.jar**) from Oracle's web page.
- If the database is a Microsoft SQL Server database, use the bundled Microsoft JDBC driver (**sqljdbc4.jar**).
  Version 4.0 of the **sqljdc4.jar** driver introduced the new **authenticationScheme= JavaKerberos** directive, which is required.

4.16.1 Workflow Overview

The following steps are required to configure Authentication of the Spotfire Database using Kerberos. Each step is later explained in detail.

1. Create a Windows domain account for the Spotfire database
2. *Microsoft SQL Server:* Create the Spotfire database
3. *Oracle:* Create and configure the Spotfire database account to the Windows domain account
4. Install Spotfire Server
5. Install the database vendor's JDBC driver
6. Configure Kerberos for Java
7. Optional: Create a Kerberos keytab file for the Spotfire database account
8. Create a JAAS application configuration for the Spotfire database connection pool
9. Register the JAAS application configuration file with Java.
10. Connect to the Spotfire database by running the **bootstrap** command or by using the Configuration Tool.

4.16.2 Detailed Instructions

▶ Create a Windows domain account for the Spotfire database:

You must be logged in as a domain administrator, a user who is a member of the built in **Account Operators** domain group, or a user with equivalent privileges. Launch the **Active Directory Users and Computers** MMC snap-in and create a normal user account with the following properties:

- Use the same identifier in the **Full name**, **User logon name** and **User logon name (pre-Windows 2000)** fields and make sure to use only lower case characters and that there are no spaces in these fields.
- Check the **Password never expires** option.
- Clear the **User must change password at next logon** option.

- *Recommended*: Check the **Account is sensitive and cannot be delegated** option.

▶ *Microsoft SQL Server:* Create the Spotfire database:

Edit and run the **create_databases_ia.bat** script. This will create a SQL Server database account and connect it to the previously created Windows domain account. See “Prepare the Database” on page 17.

▶ *Oracle:* To create and configure the Spotfire database account to the Windows domain account:

Edit and run the **create_databases.bat** script. This will create a normal Oracle database account that authenticates with user name and password. See “Prepare the Database” on page 17.

1. Log in to the Oracle database instance with SYSDBA privileges to manage accounts.

   Example: Connecting to a database with connection identifier ORCL as sysdba

   ```
   > sqlplus sys@ORCL as sysdba
   ```

2. Alter the Spotfire database account so that it is identified externally by running the following command:

   ```
   SQL> alter user <SERVERDB_USER> identified externally as '<SERVERDB_USER>@
   <REALM>';
   ```

   Replace **<SERVERDB_USER>** and **<REALM>** with the Spotfire database account name and the Kerberos realm. Make sure to use upper case when specifying the Kerberos realm

   Example:

   ```
   SQL> alter user spotuser identified externally as 'spotuser@RESEARCH.EXAMPLE.COM';
   ```

3. Test the Kerberos-enabled Spotfire database account by launching a command prompt running as the created Windows domain account. It should now be possible to connect to the database using the following command, assuming the connection identifier is ORCL:

   ```
   > sqlplus /@ORCL
   ```

   Note: It is assumed that Kerberos authentication is already set up for the Oracle client.

▶ To install Spotfire Server:

Install Spotfire Server following the instructions in the section “Install Spotfire Server” on page 23.

▶ To install a vendor database driver:

Install a vendor database driver as described in the section “Install Database Drivers” on page 27.

To configure Kerberos for Java:

Follow the instructions in “Configure Kerberos for Java:” on page 70.

Optional: To create a keytab file for the Kerberos service account:

Option 1: Using the **ktpass.exe** command included with the **Microsoft Support Tools**

On a computer with the **Microsoft Support Tools** installed (there is no need to be logged in as a privileged user), execute the following command:

```
> ktpass /princ <database account name>@<REALM> /ptype krb5_nt_principal /
crypto rc4-hmac-nt /out spotfire-database.keytab -kvno 0 /pass *
```

Replace the **<database account name>** and **<REALM>** with the appropriate values.
Note: All values are case sensitive.

- **service account name**: The **user login name** of the Spotfire database account (written in lower case)
- **REALM**: The name of the Kerberos realm, which is the DNS domain name written in upper case
- The tool will prompt for the password of the service account. Enter the same password as when creating the Spotfire database account.

It is not critical to use the name **spotfire-database.keytab** for the keytab file, but the following instructions assumes that this name will be used.

Note: If you ever change the password of the Kerberos service account in the future, you must re-create the keytab file again.

Example: Creating a keytab file for the **spotuser** Spotfire database account in the **research.example.com** domain:

```
> ktpass /princ spotuser@RESEARCH.EXAMPLE.COM /ptype krb5_nt_principal /
crypto rc4-hmac-nt /out spotfire-database.keytab -kvno 0 /pass *
```

Finally, copy the **spotfire-database.keytab** file to the directory **<installation dir>\jdk\jre\lib\security** (Windows) or **<installation dir>/jdk/jre/lib/security** (Unix) on Spotfire Server.

Note: Since this file contains sensitive information it must be handled with care. The file must not under any circumstances be readable for unauthorized users.

Option 2: Using the **ktab.exe** command included with the bundled JDK

On the computer where Spotfire Server is installed, execute the following command:

```
> ktab -k spotfire-database.keytab -a <database account name>
```

Replace the **<database account name>** as described in Option 1 above.

It is not critical to use the name **spotfire-database.keytab** for the keytab file, but the following instructions assumes that this name will be used.

The tool will prompt for the password of the service account. Enter the same password as when creating the Spotfire database account.

Note: If you ever change the password of the Kerberos service account in the future, you must re-create the keytab file again.

Verify the created keytab by running the klist and kinit utilities:

```
> klist -k spotfire-database.keytab
> kinit -k -t spotfire-database.keytab <database account name>@<realm>
```

Example: Creating and verifying a keytab file for the **serverdb_user** Spotfire database account in the **research.example.com** domain:

```
> ktab -k spotfire-database.keytab -a serverdb_user
> klist -k spotfire-database.keytab
> kinit -k -t spotfire-database.keytab serverdb_user@RESEARCH.EXAMPLE.COM
```

Finally, copy the **spotfire-database.keytab** file to the directory **<installation dir>\jdk\jre\lib\security** (Windows) or **<installation dir>/jdk/jre/lib/security** (Unix) on Spotfire Server.

Note: Since this file contains sensitive information it must be handled with care. The file must under any circumstances not be readable for unauthorized users.

Option 3: Using the **ktutil** command on Linux.

First, make sure that Kerberos is installed on the Linux host where Spotfire Server is installed. These instructions assume that the tools ktutil, klist and kinit are available on the Linux host.

Start the ktutil tool by invoking it from the command line without any arguments and execute the commands below:

```
> ktutil

ktutil:  add_entry -password -p <database account name> -k 0 -e rc4-hmac

Password for <database account name>:

ktutil:  write_kt spotfire-database.keytab

ktutil:  quit
```

Replace **<database account name>** with the actual name of the account as described in Option 1 above.

It is not critical to use the name **spotfire-database.keytab** for the keytab file, but the following instructions assumes that this name will be used.

The tool will prompt for the password of the service account. Enter the same password as when creating the Spotfire database account.

Note: If you ever change the password of the Kerberos service account in the future, you must re-create the keytab file again.

Verify the created keytab by running the klist and kinit utilities:

```
> klist -k spotfire-database.keytab
> kinit -k -t spotfire-database.keytab <database account name>@<realm>
```

Example: Creating and verifying a keytab file for the serverdb_user Spotfire database account in the research.example.com domain:

```
> ktutil

ktutil:  add_entry -password -p serverdb_user -k 0 -e rc4-hmac

Password for serverdb_user:

ktutil:  write_kt spotfire-database.keytab

ktutil:  quit

> klist -k spotfire-database.keytab

> kinit -k -t spotfire-database.keytab serverdb_user@RESEARCH.EXAMPLE.COM
```

Finally, copy the **spotfire-database.keytab** file to the directory **<installation dir>/jdk/jre/lib/security**.

Note: Since this file contains sensitive information it must be handled with care. The file must under any circumstances not be readable for unauthorized users.

To create a JAAS application configuration for the Spotfire database connection pool:

Create the file **<installation dir>\jdk\jre\lib\security\spotfire-database.login** (Windows) or **<installation dir>/jdk/jre/lib/security/spotfire-database.login** (Unix) and populate it with one of the options shown below.

- To acquire a Kerberos ticket using a keytab file, select alternative 1
- To acquire a Kerberos ticket using a user name and a password, select alternative 2.
- To acquire a Kerberos ticket using the identity of the account running Spotfire Server process, select alternative 3.

Note: Regardless if you select alternative 1, 2 or 3 below, save the file as **spotfire-database.login**.

1 To acquire a Kerberos ticket using a keytab file:

Replace **<service account name>** and **<realm>** with the name of the Spotfire database account and the Kerberos realm. Make sure to use lowercase letters for the account name and uppercase for the realm name.

```
DatabaseKerberos
{
  com.sun.security.auth.module.Krb5LoginModule
    required
    debug=true
    storeKey=true
    useKeyTab=true
    keyTab="${java.home}/lib/security/spotfire-database.keytab"
    principal="<SERVERDB_USER>@<REALM>";
};
```

2 To acquire a Kerberos ticket using a user name and a password:

Replace **<service account name>** and **<password>** with the name and the password of the Spotfire database account.

```
DatabaseKerberos
{
  com.sun.security.auth.module.Krb5LoginModule
    required
    debug=true
    storeKey=true
    useKeyTab=false
    doNotPrompt=false;
};
```

3 To acquire a Kerberos ticket using the identity of the account running Spotfire Server process:

To make it possible to login to the Spotfire database as the user currently running the server, the connection pool must be able to acquire the initial Ticket-Granting-Ticket (TGT) from the Spotfire Server's host's native Ticket Cache.

On modern Windows operating systems, the TGT session key cannot be exported unless the following registry key is modified:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\
Parameters]"allowtgtsessionkey"=dword:00000001
```

```
DatabaseKerberos
{
  com.sun.security.auth.module.Krb5LoginModule
    required
    debug=true
    storeKey=true
    useTicketCache=true
    doNotPrompt=false;
};
```

▶ To register the JAAS application configuration file with Java:

The **spotfire-database.login** file that was created above must be registered in Java.

Open the file **<installation dir>/jdk/jre/lib/security/java.security** in a text editor and add the following lines at the end of the file:

```
# Register Java Authentication & Authorization Services (JAAS)
configurations
login.config.url.1=file:${java.home}/lib/security/
spotfire-database.login
```

Configure the database connection for Spotfire Server:

Execute the **bootstrap** command (page 195). If the bootstrap command below executes successfully, the database connection is correctly established after using Kerberos authentication.

Note: It makes no sense to configure a specific Configuration Tool password when using a connection pool that logs in to Kerberos using a keytab file or retrieving the Ticket-Granting Ticket (TGT) from the native ticket cache. However, it does have a function in the scenario with a connection pool that logs in to Kerberos using a user name and password.

Oracle

To bootstrap Spotfire Server, execute the bootstrap command:

```
> config bootstrap --test --driver-class=oracle.jdbc.OracleDriver --database-url=<database
url> --kerberos-login-context=DatabaseKerberos -Coracle.net.authentication_services=
(KERBEROS5)
```

Replace **<database url>** with the JDBC connection URL. When using a user name and a password to request the Kerberos ticket, make sure to also specify the **-username** and **-password** arguments.

Oracle example:

```
> config bootstrap --test --driver-class=oracle.jdbc.OracleDriver --database-url=
jdbc:oracle:thin:@research.example.com:1521:orcl --kerberos-login-context=
DatabaseKerberos -Coracle.net.authentication_services=(KERBEROS5)
```

Microsoft SQL Server

To bootstrap Spotfire Server, execute the bootstrap command:

```
> config bootstrap --test --driver-class=com.microsoft.sqlserver.jdbc.SQLServerDriver
--database-url=<database url> --kerberos-login-context=DatabaseKerberos
```

Replace **<database url>** with the JDBC connection URL. This URL must include **;integratedSecurity=true;authenticationScheme=JavaKerberos** options.

Microsoft SQL Server example:

```
> config bootstrap --test --driver-class=com.microsoft.sqlserver.jdbc.SQLServerDriver
--database-url=jdbc:sqlserver://db.research.example.com:1433;DatabaseName=
spotfire_server;integratedSecurity=true;authenticationScheme=JavaKerberos
--kerberos-login-context=DatabaseKerberos
```

4.17 Exporting and Importing Configuration Files

Configurations are stored as xml-files in the database. The **configuration.xml** file can be viewed, edited and sent to support for inspection. To do this, **export** the file from the database. When you **import** the **configuration.xml** file to the database, it is imported and also set as the active configuration.

The configuration settings can be exported to file for backup purposes, or to be imported into another cluster to set up several clusters with similar settings.

Some configuration properties in the Spotfire system are rarely used and can not be set using commands. To use such configuration properties, export the configuration to an XML file, edit it and import it back into the Spotfire database. This procedure can also be used to configure features that require complex commands, for example several authentication options.

4.18 Configuring a Specific Directory for Library Import and Export

For performance and security reasons, Library Import and Export Operations can only be performed from a specific directory on Spotfire Server. If this for some reason is inconvenient, it can be altered.

Files that are to be imported into the Spotfire Library by Spotfire Administrators need to be placed in a specific directory in order to be imported. By default this is:
<server installation directory>/tomcat/application-data/library.
For most intents and purposes, this setting does not need to be changed.

Use the command “config-import-export-directory” on page 211 in the Library group.

4.19 Attachment Manager

The Attachment Manager is used when large amounts of data are transferred between Spotfire Clients, Spotfire Web Players or web browsers and Spotfire Server. The Library uses the Attachment Manager to transfer the actual content when downloading or saving items, and Information Services uses it for downloading the result of an information link execution.

The Attachment Manager can cache Library Items and Information Links. This Caching is enabled by default, but can be completely disabled using the **library.content-caching-enabled** and **information-services.result-caching** settings. For Library Items, the cache will never return any stale data and they are cached by default. For Information Links, each link has to be explicitly set as "cacheable" when it is created. It is the responsibility of the user that creates the information link not to mark it as “cacheable” if the result may depend on things outside the control of Information Services, such as database row level security.

The cache is encrypted by default and the size is limited in terms of disk usage. If items are to be evicted, a least recently used-algorithm is used. Library Items have higher priority and will always be evicted after Information Links.

Attachments in the cache will time out and be removed after a configurable amount of time. Default values for the cache expiration time and the max cache size are 24 hours (68400 seconds) and 10 GB (10240 MB).

5 Administration

Before installing Spotfire clients on end user machines, administrative tasks such as setting up software updates, setting up users groups, deploying a Spotfire distribution and packages, and management of licenses and setting of preferences have to be performed. There are a number of tools for administrative tasks:

- **Administration Console**: **http[s]://<server>:<server port>/spotfire**
  A web application for users, groups, and deployments management.
  The manual is available in the **Spotfire Deployment** distribution.
- **Server Logs and Diagnostics**: **http[s]://<server>:<server port>/spotfire**
  See “Server Logs and Diagnostics” on page 173.
- **Administration Manager**: TIBCO Spotfire - **Tools > Administration Manager**
  A tool in Spotfire for licenses and preferences management.
- **Library Administration**: TIBCO Spotfire - **Tools > Library Administration**
  A tool for Spotfire Library management, including import and export.

Note: Javascript must be enabled in your web browser for the Administration Console to launch.

5.1 The First Administration Process

1. **Start Spotfire Server service**
   The server must be started before you can access the Administration Console.
   See “Start and Stop Spotfire Server” on page 110.
2. **Define users and g**
   See “Manage Users and Groups” on page 112.
3. **Deploy client packages to Spotfire Server**
   A Spotfire distribution and usually additional packages are required not only for users to be able to run Spotfire, but also for the following administration steps.
   See Section 5.3.1 on page 113.
4. **Install Spotfire for the Spotfire Administrator to use**
   See “Install Spotfire for Spotfire Administrator Usage” on page 114.
5. **Assign licenses and define preferences**
   See “Manage Licenses and Preferences” on page 114.
6. **Install Spotfire clients on end users’ machines**
   See “Install Spotfire Clients on End Users’ Machines” on page 114.
7. *Recommended*: **Enable use of the data functions**
   See “Enable Date Functions Usage” on page 114.

   To use these data functions, Spotfire® Statistics Services needs to be installed and configured as well. The data functions provide Spotfire users immediately useful analytic functionality, as well as detailed and flexible templates to help users develop their own data functions more quickly and easily. Installing these data functions will make these features potentially available to your users.
8. *Optional*: **Enable use of the demo database**
   See “Enable Demo Database Usage” on page 116.

5.2 Start and Stop Spotfire Server

5.2.1 Windows

On a Windows machine the procedure differs depending on if the Windows service is created or not, and if Spotfire Server is run as a Windows Domain user. When started, this application writes log files to the **<installation dir>/tomcat/logs** directory. The procedure for stopping the server is also described.

Troubleshooting

To verify that the server can be started, launch a command prompt, go to **<installation dir>/tomcat/**bin and run the command **catalina.bat run** or **catalina.sh run**. Server debug info will be written to the console. Launch a browser and go to the Spotfire server start page: **http://<hostname>:<port>/spotfire**. Select **Open Logs and Diagnostics** to verify that authentication works and you are an administrator user.

that memory settings are different from when running as a Windows service.

5.2.1.1 Windows, Service Exists

Click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Locate the service called *TIBCO Spotfire Server 6.5* and start or stop it.

The Spotfire Server Windows service can also be started/stopped from a command prompt (you need to be logged in as an Administrator):

- **net start Tss601**
- **net stop Tss601**

Note: If you need to reinstall the Windows service after installation, run the bat file **<installation dir>/tomcat/bin/service.bat** with the argument install: **<installation dir>/tomcat/bin/service.bat install**. You can also run this script with the argument **remove** to remove the Windows service: **<installation dir>/tomcat/bin/service.bat remove**

5.2.1.2 Windows, No Service

If you did not install a Windows service you must start Spotfire Server manually.

1 Log in to the Spotfire Server machine as an administrator.

2 Start a command prompt.

3 Navigate to the folder **<installation dir>/tomcat/bin**

4 Run the **startuptomcat.bat** file.

Response: Spotfire Server starts. The server will stop running if you close the command prompt or log out of the machine.

5.2.1.3 Windows, Service Exists, Integrated Authentication for SQL Server

If your database server uses Windows Integrated Authentication, your Spotfire Server must run as a Windows Domain user that has permissions to use the Spotfire database.

1 Click **Start > Control Panel** > **Administrative Tools** > **Services**. Locate the service called *TIBCO Spotfire Server 6.5*. Double-click it to open its *Properties* dialog.

2 Select the **Log On** tab.

3 Select the **This account** radio button and enter the user credentials of the Domain User set up with the database preparation script **create_databases_ia.bat**.
Click **OK**.

4 Start or stop the service.

5.2.1.4 Windows, No Service, Integrated Authentication for SQL Server

If your database server uses Windows Integrated Authentication, your Spotfire Server must run as a Windows Domain user that has permissions to use the Spotfire database.

1 Log in to the Spotfire Server machine as the Domain User set up with the database preparation script **create_databases_ia.bat**.

2 Start a command prompt.

3 Go to the folder **<installation dir>/tomcat/bin**

4 Run the **startuptomcat.bat** file

Response: Spotfire Server starts. The server will stop running if you close the command prompt or log out of the machine.

5.2.2 Linux

For Red Hat and SUSE systems the service starts on system startup.

Note: Only a user with root user privileges can start and stop the server.

▶ Manual Spotfire Server Start:

1 Log in as **root** or run with **sudo -s**.

2 Run the command **/etc/init.d/tss start**.

▶ Manual Spotfire Server Stop:

1 Log in as **root** or run with **sudo -s**.

2 Run the command **/etc/init.d/tss stop**.

5.2.3 Solaris

To start Spotfire Server on reboot:

To configure Spotfire Server to start automatically when the Solaris machine is rebooted, run the **install_startup_script.sh** script:

1 Log in as **root**.

<u>Comment:</u> In order to have a service automatically start at reboot you must be **root**. No other user can do this.

2 Navigate to the **<installation dir>/tomcat**.

3 Execute the file **install_startup_script.sh**.

<u>Response:</u> Spotfire Server will start automatically after each machine reboot.

To start or stop the server right now, run either the script **/etc/init.d/tss start** or **/etc/init.d/tss stop**

Note: The **install_startup_script.sh** script copies the script **tss**, also located in **<installation dir>/tomcat**, to the **/etc/init.d** directory and places symbolic links to this scripts in the appropriate runlevel directories (for example **/etc/rc2.d**).

To start Spotfire Server in a terminal:

If you wish to run Spotfire Server in a terminal, then execute the command **startup.sh** located in the directory **<installation dir>/tomcat/bin** with the user who installed Spotfire Server.

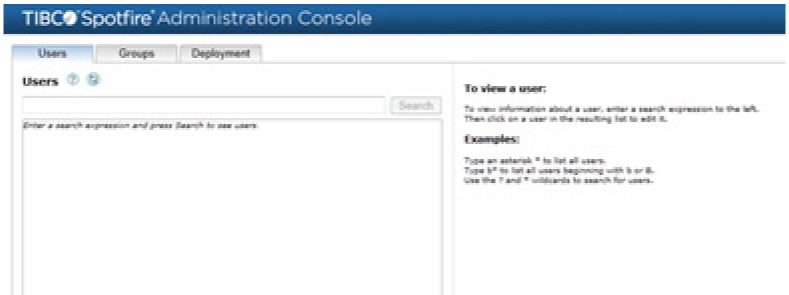
<u>Response:</u> Spotfire Server starts.

5.3 Manage Users and Groups

Spotfire licenses control which software features are available to different users. Licenses are set for groups. If you are using an external mechanism to handle users and groups, you can create Spotfire groups and put the external groups into the Spotfire groups.

Users, groups, and deployments can be created, deleted, and managed in the Administration Console. You can, for example, set new passwords, administer group membership of users, assign primary groups, and assign a deployment area to groups. See the Spotfire Administration Console online Help for more information.

http[s]://<server>:<server port>/spotfire



Enter a search expression and press Search to see users.

To view a user:

To view information about a user, enter a search expression to the left. Then click on a user in the resulting list to edit it.

Examples:

Type an asterisk * to list all users.
Type b* to list all users beginning with b or B.
Use the ? and * wildcards to search for users.

5.3.1 Prepare Spotfire for Software Updates (Deploy Spotfire)

When Spotfire users log in to Spotfire Server from a Spotfire client, the version of the client software is compared to the software on the server (the packages deployed to the server). A deployment must exist on the server for the user to be able to use Spotfire. The deployment is a bundle of software packages and licenses that is uploaded to the Spotfire database.

A Spotfire deployment is created by uploading a Spotfire Deployment distribution, and possibly additional packages, to Spotfire Server using the **Administration Console**.

A distribution file (.sdn file) is made up of package files (.spk files) that may have dependencies on one another. The **Spotfire.Dxp.sdn** distribution is the most frequently encountered example of a distribution file. It is located in the Spotfire deployment kit that you download from the TIBCO web site.

▶ To locate the Spotfire distribution file:

1. In the software downloads section of the TIBCO web site, go to **TIBCO Spotfire Server 6.5.0**, and then click **TIBCO Spotfire Deployment Kit Software 6.5.0**.
2. Download and extract the files that are in **TIB_sfire_deploy_6.5.0_win.zip**.
3. In the extracted files, double-click the **Products** folder, and then the **TIBCO Spotfire Distribution** folder.

   You will see the **Spotfire.Dxp.sdn** file

   For comprehensive help about how to deploy packages, see the *TIBCO Spotfire Server Deployment and Administration Manual*, which is available in the **Documentation** folder of the same deployment kit.

5.4 Install Spotfire for Spotfire Administrator Usage

Some procedures have to be performed on a Spotfire Client. Install a Spotfire client on a computer and log in using the administrator user you have assigned to verify that the deployment you have created works, and to get access to the Administration Manager and the Library Administration tools.

5.4.1 Manage Licenses and Preferences

To assign licenses, start Spotfire, select **Tools > Administration Manager**, and select the **Groups and Licenses** tab. See the *TIBCO Spotfire Deployment and Administration Manual* for details.

5.5 Install Spotfire Clients on End Users' Machines

When all the tasks described above are finished, the Spotfire client can be installed on each Spotfire user's computer. See the *TIBCO Spotfire Deployment and Administration Manual* for comprehensive instructions on how to do this.

You must also either start each client, go into the **Manage Servers** dialog, and add the Server URL, or distribute this URL to all users that they may enter it themselves.

5.6 Enable Date Functions Usage

Spotfire 6.5 features pre-packaged predictive analytic methods in the form of data functions. In order to leverage these data functions, Spotfire Statistics Services needs to be installed and configured. The data functions immediately provide useful analytic functionality to Spotfire users. They also provide detailed and flexible templates to help users develop their own data functions more quickly and easily. Installing the data functions will make these features potentially available to your users.

To make the data functions available to users, locate the **datafunctions** folder in Spotfire Server installation kit. Copy the file **datafunctions.part0.zip** (this ZIP file contains the data functions and analysis files) to the configured library Import/Export folder. See "Configuring a Specific Directory for Library Import and Export" on page 108.

When this file is in place, a Spotfire Administrator or a Library Administrator can import these files to the Library.

▸ To Import the data function files into the Library

1 Start Spotfire and log in as a Spotfire Admin.

2 Select **Tools** > **Library Administration**.

3 Navigate to the Library folder where you want to import the demo files. For instance, you can create a folder called **DataFunctions**.

4 Select **Import**.

5 Click **Browse** and select the file **datafunctions.part0.zip**.

6 Click **OK**.

7 Click **OK**.

8 Click **Close** when the dialog states Import Done.

5.7 Enable Geocoding Tables for Map Chart

To display data on a map, the data needs to be geocoded. Geocoding in Spotfire is the process of using some type of identifiers in a data table and matching those to similar identifiers in another set of data tables (a geocoding hierarchy) which contains latitude/longitude coordinates or geographic features. These coordinates or features are then used for correctly positioning the data in a map context.

Spotfire 6.5 features pre-packaged geocoding hierarchies, so-called geocoding tables.

To make the geocoding tables available to users, locate the **geoanalytics** folder in the Spotfire Server installation kit. Copy the file **geoanalytics.part0.zip** to the configured library Import/Export folder. See “Configuring a Specific Directory for Library Import and Export” on page 108.

When this file is in place, a Spotfire Administrator or Library Administrator can import these files to the Library.

▶ To import the geocoding tables into the Library

1 Start Spotfire and log in as a Spotfire Admin.

2 Select **Tools** > **Library Administration**.

3 Navigate to the Library folder where you want to import the demo files. For instance, you can create a folder called **GeoAnalytics**.

4 Select **Import**.

5 Click **Browse** and select the file **geoanalytics.part0.zip**.

6 Click **OK**.

7 Click **OK**.

Click **Close** when the dialog states Import Done.

5.8 Enable Demo Database Usage

To make the demo database available to users, locate the folder called **demodata** in the Spotfire Server installation kit. Select the **oracle** or **mssql** sub-folder depending on your database server, and copy the file within this folder, **demo.part0.zip** to a shared disk location. See “Configuring a Specific Directory for Library Import and Export” on page 108. This ZIP file contains analysis files and an information model that links to the demo data.

When this file is in place, a Spotfire Administrator or Library Administrator can import these files to the Library.

▶ To Import the demo files into the Library

1. Start Spotfire and log in as a Spotfire Admin.
2. Select **Tools** > **Library Administration**.
3. Navigate to the Library folder where you want to import the demo files. For instance, you can create a folder called **Demo**.
4. Select **Import**.
5. Click **Browse** and select the file **demo.part0.zip**.
6. Click **OK**.
7. Click **OK**.
8. Click **Close** when the dialog states Import Done.

A template data source will also be imported with this file. You must use the Information Designer to edit this and supply the URL and login information of your database server for it to work. See the Information Designer Help for more information about how to set up and edit data sources.

6 Monitoring

Spotfire Server can be monitored. Reasons for monitoring include detecting problems with the server itself, problems with external systems such as databases and LDAP servers, network problems, misconfigured clients, and in some cases malicious behavior. The purpose is typically to reduce downtime, detect and fix problems before users notice them, and eliminate performance bottle necks.

Spotfire Server can be monitored using TIBCO Hawk® or any other Java Management Extensions (JMX) compliant monitoring tool, like JConsole, a part of the Java SDK which is bundled with Spotfire Server. JMX is a Java framework for monitoring and managing applications and devices. It is part of the Java Platform Standard Edition since version 5.0.

See “Action Logs and System Monitoring” on page 121 for information about how to log actions running on Spotfire Server, and also events from Spotfire, Spotfire Web Player, and Spotfire Automation Services.

6.1 Instrumentation

JMX consists of three levels:

1. Instrumentation level: Provides monitoring information and management operations.
2. Agent level: Server that provides applications access to the instrumentation level.
3. Remote Management level: Connectors and adaptors providing access to the agent.

Spotfire Server runs within the Tomcat application server, which provides the basic functionality needed, the server (Agent level), and a Java Remote Method Invocation (Java RMI) connector (Remote Management level).

Tomcat provides a rich instrumentation set for monitoring and managing the application server. For example, it monitors Tomcat configuration parameters and basic usage statistics. The Java shipped with Spotfire Server is also heavily instrumented using JMX, providing information about CPU and memory usage, garbage collection, and thread pools.

Spotfire Server has been instrumented with the following measures:

Note: Also see “Action Logs and System Monitoring” on page 121.

Server

- Server address (IP)
- Server hostname
- Server version
- Date and time the server was started
- Uptime time since the server was started, both as a formatted string and in milliseconds since January 1, 1970, 00:00:00 GMT

Logging

- Current log configuration file (configurable)
- Available log configuration files (read only)
  Lists all log configuration files in **<installation dir>\tomcat\webapps\spotfire\WEB-INF**
- Number of logging events on warn, error, and fatal levels

Logger

There may be several of these or none at all, depending on the log configuration.

- Log appender name
- Notifications: Outputs all log statements from a configured log4j appender as JMX notifications

Server Metrics

- Number of attachments on the server
- Number of running Information Services jobs
- Number of authenticated HTTP sessions

HTTP Status Codes

- Number of HTTP response codes representing client or server errors, meaning the 4xx and 5xx ranges returned from the server.
  Note: Responses in these series may be common, even in a system that works well.

Data Source

One entry for each currently running data source on the server, including the server's own data source:

- Name
- URL
- Configured minimum number of connections
- Configured maximum number of connections
- Current number of active connections
- Current number of idle connections
- The maximum number of concurrently active connections seen

6.2 Configuration

Because sensitive information may be provided through JMX, and Java, Tomcat, and Spotfire Server provide some management capabilities, it is important to restrict access. The JMX RMI connector is disabled by default; the administrator must enable it. Also consider the authentication, authorization, and encryption security features.

Authentication

Spotfire Server solution applies the existing database authentication mechanism using a separate database table. Passwords are hashed and the same principals may be used across an entire Spotfire Server cluster.

Authentication is enabled by default.

Authorization

Each user has either read, or read and write, permissions. This means that the user can either only read attribute values or, in addition, read and modify the attributes if they are writable.

Authorization is enabled by default. Authorization only works with the default authentication implementation.

JMX accounts and credentials are separated from Spotfire accounts and credentials. The JMX accounts are only used for monitoring, since ordinary Spotfire login does not work.

Encryption

The RMI connector can be configured to encrypt the traffic using SSL. This is recommended since user names and passwords are otherwise transmitted in plain text.

SSL is not enabled by default. It requires a certificate.

Firewalls

A firewall can be configured to allow traffic to the desired ports. By default the RMI registry and the RMI connector share a common port (1099) to simplify firewall configuration.

JMX Configuration Commands

The following commands are used to configure and administrate JMX access to the monitoring component.

config-jmx	Configures the JMX RMI connector
create-jmx-user	Creates a new JMX user account
delete-jmx-user	Deletes a JMX user
list-jmx-users	Lists all JMX users

6.3 Example

This is the procedure to set up JMX monitoring using JConsole. This example does not use SSL:

1 Run the **config-jmx** command: **config config-jmx --enabled=true**

2 Import the configuration: **config import-config --comment=”Enabling JMX” configuration.xml**

3 Provide the configuration tool password.

4 Create a JMX user: **config create-jmx-user --username=MyJMXUser**

5 Provide the MyJMXUser password.

6 Provide the configuration tool password to write the user and password to database.

7 Restart Spotfire Server.

8 Launch the JConsole application. In a default Windows installation: **C:\tibco\tss\6.5\jdk\bin\jconsole.exe**

9 In the *JConsole New Connection* dialog, select **Remote Process**, enter the **<hostname>:1099**, and provide the JMX user name and password.

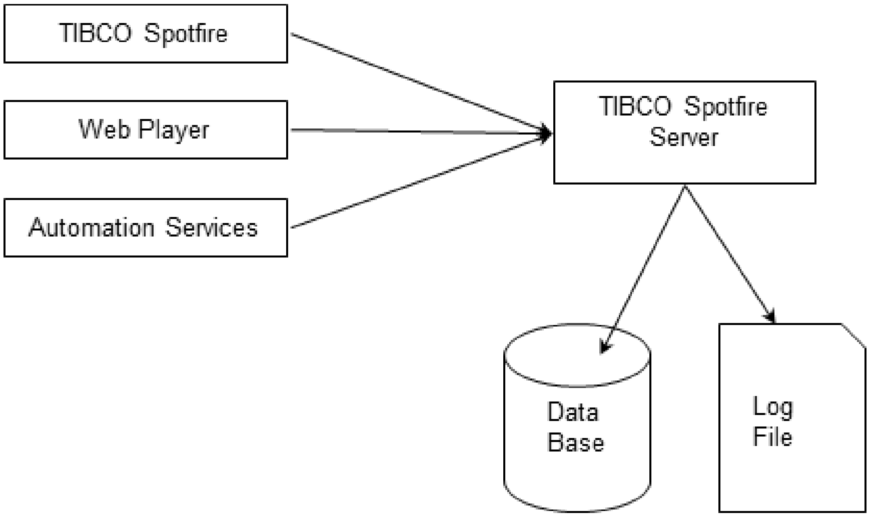
Comment: To view the Spotfire specifics, see the MBeans tab and the **com.spotfire.server** domain.

7 Action Logs and System Monitoring

7.1 Introduction

The action logs feature collects information about what the users are doing and the system monitoring collects information on the performance of Spotfire Server and the Web Player Server. As these different logging events are written to the same file or database, it is possible to correlate the usage with the system performance.

The log events can be written to files, to a database, or to both. In contrast to the other log files, these log files will not be pruned; instead a new file will be created every day, thus some extra administration is needed to ensure that there is room in the file system. For the database logging there is an option to automatically remove entries which are older than a certain number of hours.



It is possible to analyze the gathered data using Spotfire. For the database there is an Information Model and an analysis file which can be used to start analyzing usage patterns. With the collected data it should be possible to answer many more questions on how the system is used. The action logs and system monitoring feature is turned off by default.

Action Logs

The action logs feature collects information about what the users are doing, for example if a user opens a file from the library, when a user logs in, etc. It will answer questions on “who did what”, but not static questions like “who can do what”, but you see when someone gives more rights to someone. It does not only log actions running on the server, but also events from Spotfire, Spotfire Web Player, and Spotfire Automation Services. All events are collected on Spotfire Server. The events that do not originate from the server are sent to Spotfire Server through a web service.

Note: The web service must be enabled and configured for these other events to be logged.

System Monitoring

The system monitoring saves information on the performance of Spotfire Server and the Spotfire Web Player Server in the same database or files as the action logs.

In contrast to the action logs, where events are logged when an action is performed, the system monitoring collects information at regular intervals. To reduce the number of measurements in the database over time, measurements older than a specified amount of time will be replaced with average, minimum, and maximum values for a period of time. The general pruning for the database will also affect the monitoring values. If you log to file, no pruning or averaging will be done.

7.2 What Is Being Logged

To decide if you want to enable the action logs and system monitoring feature, it is instructive to see what information it provides. The log points are separated into different categories; for the categories there are different actions, for example when a user changes their password it belongs to the "admin" category and the action is "change_passwd". For every log point there are some generic fields which are shared among log points, these are:

logged_time; machine; user_name; original_time; original_ip; category;action; success; session_id,

And then there are some specific measures for every log point, for example when we log that a user changes password we log **uName**, meaning the user name.

The generic fields are described below.

logged_time	The time the event was logged.
machine	The machine that did the logging.
user_name	The name of the authenticated user performing the logged action.
original_time	The time the event originally was created. This might differ from the logged time because it can take time for the log event to be written.
original_ip	Where the call originates. We will check on TCP level, so it might be a proxy that shows up here.
category	The category of the event, for example admin.
action	The action within the category, for example change_passwd.
success	Tells if the operation succeeded or not. For 6.5 the emphasis is on the successful operations.
session_id	A (unique) id for the session.

Apart from these there are some variable fields. In the database these will fill out **id1, id2, arg1, arg2** etc. For the database there are also database views which will have the generic column names altered to the ones in the table below. For the change password there is a specific view, which for Oracle is defined as:

```
CREATE OR REPLACE VIEW ADMIN_CHANGE_PASSWD  AS SELECT LOGGED_TIME,
MACHINE, USER_NAME, ORIGINAL_TIME, ORIGINAL_IP, SUCCESS, SESSION_ID,
ID1 AS UNAME FROM ACTIONLOG WHERE LOG_CATEGORY = 'admin' AND
LOG_ACTION = 'change_passwd'
```

The following log points exist. If the category has the suffix **_pro** it means that the operation is coming from Spotfire, **_wp** means that it is coming from the Spotfire Web Player, and **_as** means that it is coming from the Spotfire Automation Services. The operations without a suffix all originate on the server. It is possible to configure the monitor so that only certain categories are logged.

7.2.1 Action Logs

Category	Action	id1	id2	arg1	arg2	arg3	arg4
admin	change_passwd	uName					
admin	create_group	gName	displayName	email			
admin	create_user	uName	displayName	email			
admin	group_add_member	name	gName	sort	groupingId		
admin	group_remove_member	name	gName	sort	groupingId		
admin	remove_license	gName	licenseName				
admin	remove_principal	name	sort	groupingId			
admin	rename_principal	oldName	newName	sort			
admin	set_license	gName	licenseName	excludingFunction			
admin	set_preference	name	prefType	category	id		
analysis_wp	set_page	analysis_id	path	pageName			
auth	impersonate	uName					
auth	login	clientType	clientVer	displayName	email		
auth	logout	uName					
auth_as	login	uName					
auth_as	logout	uName					
auth_pro	login	uName					
auth_pro	logout	uName					
auth_wp	login	uName					
auth_wp	logout	uName					
dat_con_pro	create_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	
dat_con_pro	create_source	libraryId	libraryPath	dataSourceType	dataSourceInformation		

dat_con_pro	get_data	libraryId	libraryPath	dataSourceType	dataSourceInformation	internalQuery	numRows
				arg5: duration	**arg6**: externalQuery		
dat_con_pro	load_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	
dat_con_pro	load_source	libraryId	libraryPath	dataSourceType	dataSourceInformation		
dat_con_pro	synch_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	
dat_con_pro	update_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	
dat_con_pro	update_source	libraryId	libraryPath	dataSourceType	dataSourceInformation		
dat_con_wp	create_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	
dat_con_wp	create_source	libraryId	libraryPath	dataSourceType	dataSourceInformation		
dat_con_wp	get_data	libraryId	libraryPath	dataSourceType	dataSourceInformation	internalQuery	numRows
				arg5: duration	**arg6**: externalQuery		
dat_con_wp	load_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	
dat_con_wp	load_source	libraryId	libraryPath	dataSourceType	dataSourceInformation		
dat_con_wp	synch_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	
dat_con_wp	update_connection	libraryId	libraryPath	dataSourceType	dataSourceInformation	dataSourceLibraryId	

dat_con_wp	update_source	libraryId	libraryPath	dataSourceType	dataSourceInformation		
datafunction_pro	execute	unused	path	params	duration		
datafunction_wp	execute	unused	path	params	duration		
datasource_pro	execute	unused	path	title	params	duration	numRows
datasource_wp	execute	unused	path	title	params	duration	numRows
file_pro	load	unused	path				
file_wp	load	unused	path				
info_link	create_il	libraryId	path				
info_link	get_data	libraryId	path	duration	sizeb	groupingId	
info_link	load_il	libraryId	path	groupingId			
info_link	update_il	libraryId	path				
library	copy	libraryId	path	libraryType	destLibraryId	destPath	groupingId
library	create	libraryId	path	libraryType	preSize	postSize	
library	delete	libraryId	path	libraryType	groupingId		
library	export	libraryId	path	destPath	groupingId		
library	import	libraryId	path	destPath	groupingId		
library	load_content	libraryId	path	libraryType	duration	sizeb	groupingId
library	move	libraryId	path	libraryType	destLibraryId	destPath	groupingId
library	remove_perm	libraryId	path	name	sort		
library	save_content	libraryId	path	libraryType	preSize	postSize	
library	set_group_perm	libraryId	path	gName	permission	groupingId	
library	set_user_perm	libraryId	path	uName	permission	groupingId	

library_as	load	library Id	path				
library_pro	close	library Id	path				
library_pro	load	library Id	path				
library_wp	clone	library Id	path				
library_wp	close	library Id	path				
library_wp	load	library Id	path				

The different measures have the following meanings.

analysis_id	A (unique) id for the instance of the analysis.
category	The category of the preference.
clientType	The type of client is it, for example “TIBCO Spotfire Web Player”.
clientVer	The version of the client that is connecting, for example “6.5”.
dataSourceInformation	Connector-specific information about the data source. Typically the location of the database.
dataSourceLibraryId	The library identifier of the connected data source, if applicable.
dataSourceType	The type of external data source.
destLibraryId	The destination library id.
destPath	The destination library path.
displayName	The display name for a user, for example “John Smith”.
duration	The amount of time the operation/operations took (in ms).
email	The e-mail address.
excludingFunction	For licenses, this is a subfunction within a license which is not turned on.
externalQuery	The external query, as generated by the adapter.
gName	The group name.
groupingId	Operations related to the same operation can share a common groupingId. For some operations this is the same as the job-id seen in the other logs.

id	The name of the preference.
internalQuery	The Spotfire query.
libraryId	The id of the library item.
libraryPath	The library path.
licenseName	The license name.
libraryType	The type of library, for example dxp. query.
name	The name of the entity.
newName	The new name.
numRows	The number of rows returned.
oldName	The old name.
pageName	The name of the page.
params	For certain operations we do not have the exact functionality, but this information can help to decide what has happened.
path	The path.
permission	The permission.
postSize	The size afterwards (in bytes).
prefType	The type of the preference.
preSize	The size before (in bytes).
sizeb	The size (in bytes).
sort	The type it is (user or group).
title	The document title.
uName	The user name.
unused	This is currently not used.

When logging to file, **the user “john” has changed password**, can look something like:

```
2013-05-07T11:55:36.356+0200;10.100.33.227;john;2013-05-07T11:55:36.3
55+0200;0:0:0:0:0:0:0:1;admin;change_passwd;true;b549dfcf-0059-4d63-b
7d0-f710cc10a3cc;john;null
```

Another example, where a file originally opened from the library has been closed on Spotfire:

```
2013-05-07T11:55:36.356+0200;10.100.33.227;sfal;2013-04-08T16:20:14.2
03+0200;null;library_pro;close;true;22154702-8e44-4a26-a102-f1a63121f
763;4447a4f7-2c33-43f0-9ed7-edafa152969f;/Demo/Baseball Deb
```

Every log event will be placed on a new row; in the log file the semicolon is used as separator; in the database the information is placed in different columns. Some columns are generic and some columns will have different meaning depending on the category and action.

When logging to database, there is one more category “dblogging”. It has three actions:

- pruned, when things are removed as a result of the pruning action
- startup, when we are starting to log (meaning when the server is started)
- shutdown, when the server is shut down (there is a risk that this is lost if the grace period is too short, but normally it should be there. Grace period will be explained later)

7.2.2 System Monitoring

Category	Action	id1	id2	arg1	arg2	arg3	arg4
monitoring	average	measure	unused	mean	min	max	
monitoring	measurement	measure	unused	value			
monitoring _wp	average	measure	unused	mean	min	max	
monitoring _wp	counter	measure	wp_id	value	counter category	counter name	counter instanc e

wp_id is a unique id that identifies the currently running instance of the Web Player.

There are different measures(id1) for Spotfire Server and the Spotfire Web Player. They have the following meanings:

Spotfire Server

Measure	Value
cpu	Average CPU load, in percent.
mem	Heap memory used, in megabytes.
sessions	The number of authenticated HTTP sessions.

Spotfire Web Player Server

Measure	Value
available bytes	The available number of bytes on the Web Player Server.
cached docs	The number of cached documents.
cpu	Average CPU load, in percent.
disk queue	The length of the disk queue.
mem	The number of bytes used by the Web Player Server.
network	The total number of bytes transferred per second.
open docs	The number of open documents.
scheduled updates docs	The number of documents controlled by the scheduled updates feature.
uptime	The time in seconds since the Web Player Server was started.

7.3 Web Service

To be able to capture log points from Spotfire, Spotfire Web Player, and Spotfire Automation Services there is a web service. It is possible to decide that only certain categories should be logged through the web service. To ensure that no unnecessary SOAP traffic is generated, the clients will check with the server during startup for the active categories. If the feature is not enabled then no extra SOAP calls will be generated.

There are three settings on the server. If it should be turned on at all, which categories should be enabled ("all" will turn on all categories) and a regular expression to decide if logging requests should be accepted or not (".*" will accept from any host).

7.4 Log File

As an option, action logging can be directed to a file. In contrast to our other logs, a new file will be created every day. You can see in the log4j configuration files that it uses the DailyRollingFileAppender. Files will never be automatically removed; thus, if it is enabled, you need to make sure that there is room for these files. Fields are separated by a semicolon, and any semicolon in the measures will be replaced with sentence spacing. The file can be opened directly in Spotfire.

Example of a log file:

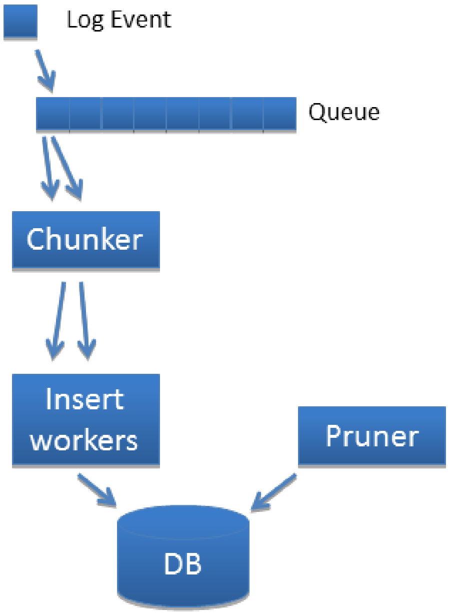
```
2013-04-08T16:15:35.062+0200;10.100.33.227;pasp;2013-04-08T16:15:35.0
61+0200;10.100.33.209;auth;login;true;259bcfcd-cd75-4757-a41e-99b06e2
8fdb1;Spotfire Web Player;6.5.0

2013-04-08T16:15:38.729+0200;10.100.33.209;pasp;2013-04-08T16:15:36.3
30+0200;::1;auth_wp;login;true;259bcfcd-cd75-4757-a41e-99b06e28fdb1;p
asp;null

2013-04-08T16:15:42.029+0200;10.100.33.227;pasp;2013-04-08T16:15:42.0
28+0200;10.100.33.209;library;load_content;true;259bcfcd-cd75-4757-a4
1e-99b06e28fdb1;4447a4f7-2c33-43f0-9ed7-edafa152969f;/PAsp/Baseball
Deb;0000000013;0000383230;null

2013-04-08T16:15:43.328+0200;10.100.33.209;pasp;2013-04-08T16:15:41.9
00+0200;::1;library_wp;load;true;259bcfcd-cd75-4757-a41e-99b06e28fdb1
;4447a4f7-2c33-43f0-9ed7-edafa152969f;/PAsp/Baseball Deb;AnalysisDxp

2013-04-08T16:15:57.720+0200;10.100.33.209;pasp;2013-04-08T16:15:56.3
67+0200;::1;library_wp;close;true;259bcfcd2013-04-08T16:16:19.185+020
0;10.100.33.209;pasp;2013-04-08T16:16:17.541+0200;::1;auth_wp;logout;
true;259bcfcd-cd75-4757-a41e-99b06e28fdb1;pasp;null

2013-04-08T16:16:21.104+0200;10.100.33.209;pasp;2013-04-08T16:16:19.9
73+0200;::1;library_wp;close;true;259bcfcd-cd75-4757-a41e-99b06e28fdb
1;4447a4f7-2c33-43f0-9ed7-edafa152969f;/PAsp/Baseball Deb

2013-04-08T16:16:28.287+0200;10.100.33.227;unknown;2013-04-08T16:16:2
8.286+0200;10.100.33.209;auth;logout;true;259bcfcd-cd75-4757-a41e-99b
06e28fdb1;pasp;null
```

The log files will show up in a subdirectory of the usual logging directory:

<installation dir>/tomcat/logs/actionlogs

7.5 Database Logging

As an option, action logging can be directed to a database. There are many configuration options available for the database logging, which will make it possible to tailor the system for your needs. To see how this functionality works it is illustrative to follow how an event is logged.



1 An event is created.

2 A check is done to see if logging is turned on.

3 A check is performed to see if this category should be logged.

4 It is fed to one or two of the loggers.

5 If file logging is enabled it will be written to the file.

6 A check is made to see if logging should be done towards the database.

7 The database logger will put the event in a fixed size queue (the size is fixed in runtime, but can be configured). It is also possible to configure the prioritization of events so that only certain events will be put in the queue if the queue is more than half full.

8 If the queue is full it can be configured to wait until there is room in the queue or wait for a configurable time.

9 The chunk worker will wait until there are a configurable number of events available or a certain configurable time has passed.

10 The chunk worker will start up an insert worker. The number of simultaneous insert workers can be configured. If the limit of simultaneous workers is reached it will wait for an insert worker to finish.

11 The insert worker will do a batch insert into the database.

As you can see there are several possibilities here to configure the system. If it is very important that everything be logged, you should block for a place in the queue.

If some elements are more important to log than others, they can be set as prioritized. This means that if the queue is more than half full, only events set as prioritized will be added to the queue. Other events will be discarded.
Note: To ensure that important elements are never discarded, you must also configure the queue to wait if it is full.

If there is a high load, you should configure many simultaneous insert workers. On the other hand if you just want to sample the system and you do not want to load a database instance, you could set the number of insert workers to a low number.

There is an optional pruner thread which, if enabled, will check every hour for events older than a configurable number of hours. The events which are older will be removed. By default the system will delete events older than 240 hours. If the value is set to 0, no pruning will take place and your DBA must administer the growth through some other means, for example by partitioning the table.

If there still are events in the queue when the server is about to be stopped, there will be an attempt to write remaining items in the queue to the database during a grace period. The grace period is also configurable.

As mentioned above, many parameters of the machinery are configurable. This should make it possible to tune the system for different environments and loads. To help tune the system there is a JMX (see “Monitoring” on page 117 for more information about JMX). This JMX bean can answer the following questions:

- How many more events can be queued? (getRemainingQueueCapacity())
- How many events are in the queue? (getCurrentQueueSize())
- How many events have tried to be logged? (getNumberOfLogged())
- How many events have not been put in the database? (getNumberOfFailedLogs())
- How many more insert workers can be started? (getCurrentNumberOfSpareWorkers())
- What is the minimum number of spare insert workers since the server was started? 0 indicates that all possible workers were started at some point. (getMinimumFreeWorkers())
- How many SQL Exceptions have been encountered? (getNumberOfSQLExceptions())
- How many items have been pruned from the database? (getNumberOfPrunedEntries())

During startup the database logger will try to connect to the database. If it fails it will try to reconnect at increasing intervals. If no database is available after the start attempts, the server will not run. Thus, if the functionality is enabled, there is another system dependency.

If you want to send information to a database, you need to run additional database scripts. These will create a new schema/database for the action logs to make it simpler to, for example, partition the data table. Everything is logged to the table “ACTION-LOG”. Then some indices are created. If you do not do searches, you can omit the indices. If you have them turned on and also have pruning, then your DBA should consider rebuilding the indices periodically. Then there are views created for categories and actions; these will help to interpret the generic columns. If you do not use the views you can omit them from the database creation script.

For the database there is also an Information Services model and an analysis file, which can be used to gain insight into the usage of the system.

7.6 Installing the Action Logs and System Monitoring Feature

By default nothing is turned on. To turn it on you need to configure it. If you want to run database logging you also need to run additional installation scripts. If you turn on database logging you can also import a library file, which will provide an information model and analysis file.

The configuration of this feature has three commands:

1 **config-action-logger** This controls whether the feature is enabled. The default is that it is turned off. If it is on, then it controls which categories should be logged, it also controls if logging should be directed towards file and/or database.

An example where all categories are enabled and we log both to file and database:

```
config-action-logger --file-logging-enabled=false
--database-logging-enabled=true
```

2 **config-action-log-database-logger** This controls the different tuning parameters of the database logger as well as the database connection information.

Example if you only want to run with the default parameters:

```
config-action-log-database-logger --database-url=
"jdbc:tibcosoftwareinc:oracle://some.oraserver.com:1521;SID=orcl"
--driver-class="tibcosoftwareinc.jdbc.oracle.OracleDriver"
--username="spotfire_actionlog" --password="xxxxx"
```

If you want to log to a database then you need to run scripts which will create a new database/schema. These are available in the installation kit in these folders:

./scripts/mssql_install/actionlog

./scripts/oracle_install/actionlog

Here the **bat** or **sh** file needs to be edited. The information is the same as for the ordinary creation scripts. For Oracle a new schema is created for the “spotfire_actionlog” user. If you want to use the information layer later then you should not change this user. For Microsoft SQL-server the database will be called “spotfire_actionlog”. If you want to use the information layer you should not change this name.

In the same folder there is a library “**logged_user_actions_ora.part0.zip**” (for Oracle) or “**logged_user_actions_mssql.part0.zip**” (for Microsoft SQL Server). This file needs to be copied to the library import folder (**<installation dir>/tomcat/application-data/**

library/) and then imported into the library using the library manager. This library export contains an information layer as well as an analysis file. To be able to use the file you need to edit the datasource with the connection information to the schema/ database. Use Information Designer and select the Datasource tab, right click on the logged_user_actions_datasource, select edit. Then edit the connection information.

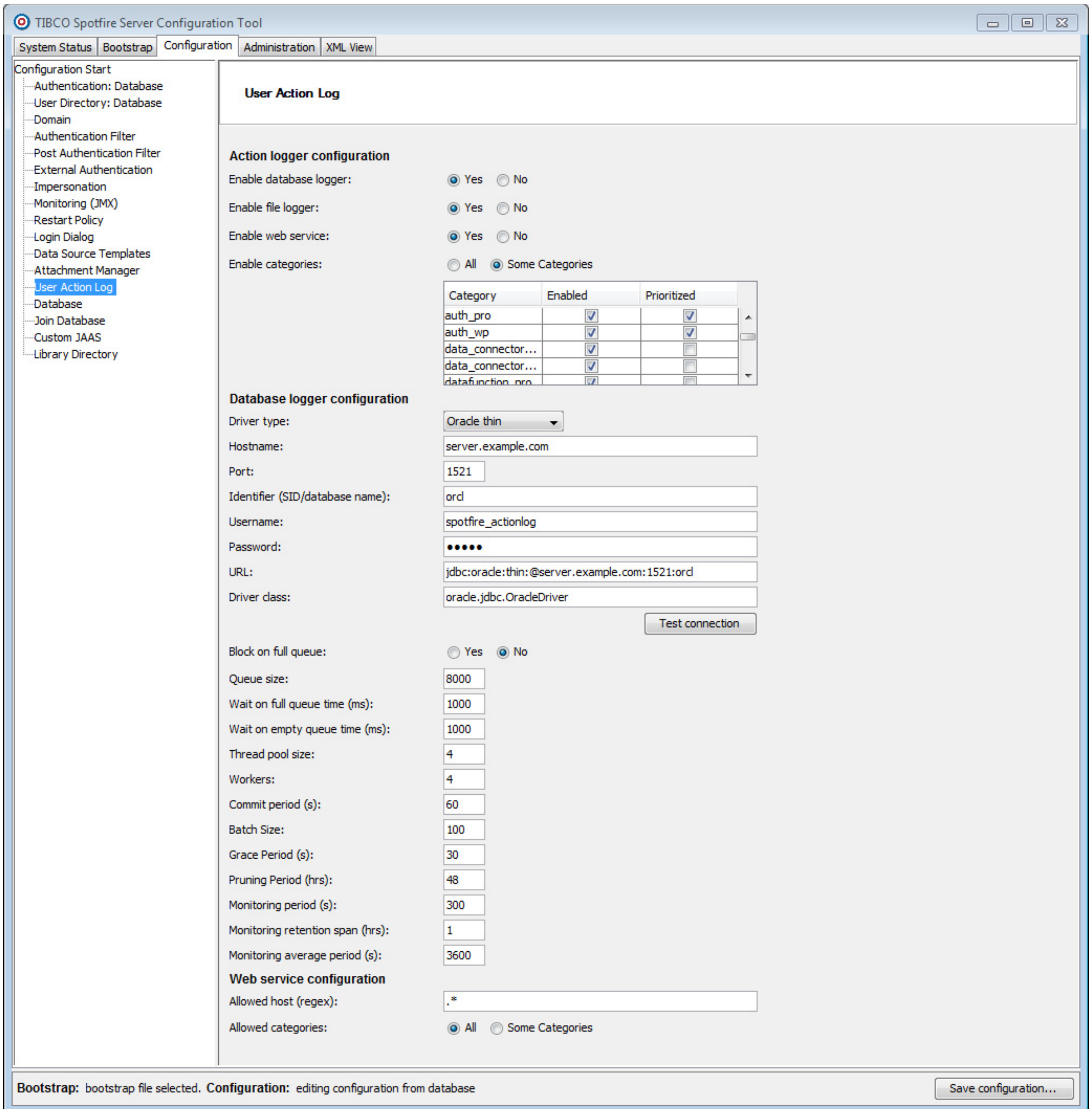
3 **config-action-log-web-service** This controls which categories are logged and also limits the clients that can log using the web service.

Example: enable all categories from all hosts:

```
config-action-log-web-service --allowedHosts=".*" --categories="all"
```

Then the configuration needs to be uploaded to the database and the server should be recycled.

It is also possible to configure the functionality through the Configuration Tool.



User Action Log

Action logger configuration

Enable database logger: ● Yes ○ No

Enable file logger: ● Yes ○ No

Enable web service: ● Yes ○ No

Enable categories: ○ All ● Some Categories

Category	Enabled	Prioritized
auth_pro	☑	☑
auth_wp	☑	☑
data_connector...	☑	□
data_connector...	☑	□
datafunction_pro	☑	□

Database logger configuration

Driver type:	Oracle thin
Hostname:	server.example.com
Port:	1521
Identifier (SID/database name):	orcl
Username:	spotfire_actionlog
Password:	•••••
URL:	jdbc:oracle:thin:@server.example.com:1521:orcl
Driver class:	oracle.jdbc.OracleDriver

Block on full queue: ○ Yes ● No

Queue size:	8000
Wait on full queue time (ms):	1000
Wait on empty queue time (ms):	1000
Thread pool size:	4
Workers:	4
Commit period (s):	60
Batch Size:	100
Grace Period (s):	30
Pruning Period (hrs):	48
Monitoring period (s):	300
Monitoring retention span (hrs):	1
Monitoring average period (s):	3600

Web service configuration

Allowed host (regex): .*

Allowed categories: ● All ○ Some Categories

Bootstrap: bootstrap file selected. **Configuration:** editing configuration from database

7.7 Some Comments

The information above about log categories, actions, and measures should not be considered as a stable API which will remain unchanged between releases. All things can change, but it is more likely that we will add more actions and add measurement columns to existing log points.

The log points represents what is happening on the system. There might be a couple of cases where what is shown in the log can feel counterintuitive, for example when using NTLM you will see more logins. If you see what is happening on the network you will see that there are actually several logins happening during a normal session. Another case is when a session dies. There is a maximum life span for a session. Here you will see an event even if the user has not actively made any operation. You can also see that there might not be a session when these events are logged, because the session has died.

If you are logging to a database then it might be a good idea to involve your DBA to regularly monitor the usage and see if indices should be rebuilt or dropped. If pruning is not turned on then manual pruning or partitioning must eventually take place.

Files from a previous release take a certain path through the code. For certain older files the clone operation on the web player might not be logged.

7.8 Upgrading Action Logs and System Monitoring

If you have been running action logging in a previous release, then logging will run out of the box, but you might not be able make full use of the new functionality.

The new functionality includes further measurements for some log points, and new measures e.g. CPU usage. Depending on which categories that were enabled earlier you might want to review these (also for the web service). If you are using the configuration tool it should be easier to choose categories since there are check boxes to select categories. If you previously had "all" selected the new categories will show up.

If you are only logging to file then there is nothing more that needs to be done.

However, if you are logging to database, there are some things to note. As before, all measures are logged to one single table "ACTIONLOG", so without any alterations your logging should continue to work and you should not lose any measurements. This "ACTIONLOG" table is the only thing required to run the logging, but as before we have some utilities that will help you to analyze the data.

There is no SQL that is run automatically during upgrade related to this logging functionality. This is to give full control to you and your DBA if you have chosen to do something advanced, e.g. partitioned the "ACTIONLOG" table.

The database scripts have basically the following functionality.

1. Create user, schema/database, after and upgrade you can continue to log to the same place so there is no need to create these anew.
2. Create the ACTIONLOG table, this table is still used and the structure is not altered.
3. Indices are created to help searches on the ACTIONLOG table. If you chose to omit the creation of the indices before and you are happy with that, then there is no need to create them this time either. With pruning enabled, the ACTIONLOG table will have both rows added and deleted so indices might benefit from being rebuilt regularly, discuss this with your DBA.
4. Views are created for the different categories and actions with column names which are more informative, with the same information as in the table in “What Is Being Logged” on page 122. The views are needed only if you use them for analysis. During an upgrade these are the only things that need to be updated in the database.

The view creation information exists in the database installation scripts, they can be found in the installation kit under

./scripts/oracle_install/actionlog

./scripts/mssql_install/actionlog

Oracle

If you are a familiar with SQL utilities it is probably fastest to log in to the schema **spotfire_actionlog** and run the SQL found in **create_actionlog_db.sql**. SQL will see if the table exists and will then only create the views.

You can also edit the .bat or .sh files. In this file remove the section which creates the tablespace and user, enter the information for: CONNECTIDENTIFIER, ACTIONDB_USER, ACTIONDB_PASSWORD, and run the script.

Microsoft SQL Server

Edit the file **create_actionlog_db.sql**

Remove the lines above “use $(ACTIONDB_NAME)” and change this line to ”use spotfire_actionlog”. The script will only create the views if the table exists.

If you are a familiar with SQL tools it is probably fastest to log in to the database **spotfire_actionlog** and run the SQL in your edited **create_actionlog_db.sql**.

You can also run the bat script. Here you need to edit the bat script. In this file remove the section “Create the Spotfire Action log database user” then enter the information where the placeholders are, for example the CONNECTIDENTIFIER, and run the script.

To help to analyze the content of the table and the views there is an information layer. This has been updated with the new views. In the same folder as the database script there is a library import file, **logged_user_actions_ora.part0.zip** (for Oracle) or **logged_user_actions_mssql.part0.zip** (for Microsoft SQL Server). This file needs to be copied to the library import folder (**<installation dir>/tomcat/application?data/library**/) and then imported into the library using the library manager. When importing this you should select to replace existing items. This library export contains an information layer as well as an example analysis file. To be able to use the file you need to edit the datasource with the connection information to the database/schema. Use Information Designer and select the **Datasource** tab, right click on **logged_user_actions_datasource**, and select **Edit**. Then edit the connection information. Check the permissions on the imported folder so that only the proper users can view the content.

7.9 Spotfire Server and the Different Databases/Schemas

There are several kinds of databases or schema that the server connects to.

- Its own database which stores all the information like preferences, library items, etc.
- The data sources to which the server makes JDBC connections to retrieve data for analysis, through Information Services. One of the possible data sources is the demo data source which can be created at the same time as the Spotfire database.
- The new action log database/schema, which is created if you want to direct the action logs to a database. It is a very simple structure with basically one table and different views which can help to analyze the content. It is separate from the Spotfire database to allow since it is a public table and you might want to handle pruning in a specific way.

8 Upgrading

There are four different upgrade scenarios for Spotfire Server:

- Upgrade to version 6.5 from a previous version of Spotfire Server, version 3.0 or higher
- Migration from Spotfire Analytics Server 10.1
- Upgrade between service pack versions
- Applying hotfixes

Each upgrade scenario and related tasks are described in separate sub-chapters.

8.1 Upgrade to 6.5 from a Previous Version of Spotfire Server

This section describes upgrades to Spotfire Server 6.5 from Spotfire Server 3.0 or higher.

If you intend to upgrade your Spotfire Server from version 3.0 or later to 6.5, use the Spotfire Server Upgrade tool. It upgrades the Spotfire database to the current version and, if selected, copies certain files from an old installation of Spotfire Server to the Spotfire Server 6.5 installation directory.

Note: After the Spotfire database is upgraded, older versions of Spotfire Server will not be able to connect to it. Therefore, stop any older Spotfire Servers connected to the Spotfire database before beginning an upgrade. If you intend to copy information from the old version, do not uninstall it until Spotfire Server 6.5 is in place.

Note: The upgrade will perform a validation of LDAP configurations. If an invalid LDAP configuration is found, the upgrade will fail. If so, go back to your previous installation, correct the error and perform the upgrade again.

Note: When upgrading from version 3.x.x or 4.5.x to 5.0 or later, a new "UTC_TIME" database view will be created for the Spotfire database. This is typically performed by the upgrade tool, but may be a manual step if the database user lacks sufficient permissions. If so, you will be given instructions by the upgrade tool and may need assistance from your DBA.

8.1.1 Install Spotfire Server

The Spotfire Server Upgrade tool is installed with Spotfire Server. First install Spotfire Server 6.5. See “Install Spotfire Server” on page 23.

Note: If you are using LDAPS, and if the CA certificate is not included in the cacert file by default, you must import the CA certificate used to issue the LDAP server's certificate according to the instructions in section “Configuring LDAPS” on page 99, **before** running the upgrade tool.

Note: Do not start or configure the newly installed server before running the upgrade tool.

8.1.2 Apply the Latest Server Hotfix

It is important that you apply the latest server hotfix before you run the upgrade tool, see “Hotfix Installation” on page 27.

8.1.3 Run the Upgrade Tool Interactively

HAVE YOU BACKED UP YOUR SPOTFIRE DATABASE?

Before you run the upgrade tool, it is very important that you have a working backup of your Spotfire database.

ARE YOU USING MICROSOFT SQL SERVER WITH WINDOWS INTEGRATED LOGIN?

If Spotfire Server is set up to authenticate with the Spotfire database using Windows Integrated Authentication, it is important that you run the upgrade tool as the same user that Spotfire Server authenticates as. Otherwise, the upgrade tool will not be able to authenticate with the database.

In the Spotfire Server 6.5 installation directory **<6.5.0 install directory>\tools\upgrade** you can find the files **upgradetool.bat** (Windows) and **upgradetool.sh** (Unix). Run the tool for your operating system from a command line prompt. It can also be launched in the last step of the Spotfire Server installer.

Note: If you are upgrading a cluster of Spotfire Servers, run the upgrade tool on each server. The Spotfire database will be updated the first time you run the upgrade tool.

The upgrade tool consists of a number of panels which will prompt you for information regarding your 3.x, 4.5, 5.0, 5.5 or 6.0 installation and your 6.5 installation.

▶ Running the upgrade tool:

1 The *Spotfire Server 6.5 Upgrade* panel is displayed. It provides a reminder to back up or clone the Spotfire database.

Click **Next**.

2 The *File Locations* panel is displayed. It provides new information and the choice to copy, or not to copy, an existing configuration. If you have file access to an old installation, you can select the **X.X.X installation** option and enter the path to its installation directory, for instance: **C:\tibco\tss\6.0** or **/opt/tss/6.0**

Note: If there are changes needed after upgrade, for example, port configuration or location of SSL certificate, manually edit the **server.xml** file, located under the **<Spotfire Server Install Dir>\tomcat\conf** folder.

If you select to not copy configuration, you will have to enter all information manually in the following panels.

Click **Next**.

3 The *Database Type and Driver* panel is displayed. If you selected to copy an existing configuration, you will not see this panel.

Specify the database and database driver you are using. If you select a database driver type that is not installed in the Spotfire 6.5 installation directory the message "The selected driver must be installed manually" will be displayed. Install the selected driver manually by placing it in the **<6.5.0 install directory>/tomcat/lib** directory and restart the upgrade tool.

Comment: If you select a database driver type that is not installed and click **next**, the *Database Drivers Not Installed* panel is displayed, informing that the upgrade tool cannot find a suitable database driver to connect to the database. This only happens if the selected database driver in the previous panel is not installed. If this occurs; click **Done** to exit the upgrade tool, install the database driver and start the upgrade tool again. If there are no database driver issues, you will not see this panel.

Click **Next** to continue.

4 The *Database Connection Information* panel is displayed. If you selected to copy configuration, the information on this panel will be populated.

- Select the **Integrated login** check box if your database server uses integrated login, like Windows authentication, to disable the *Username* and *Password* fields.
- Provide the 3.x Spotfire database **Connection string**, **Username**, and **Password**.

Click **Next**.

5 The *Additional Information* panel is displayed if upgrading from 3.x or if "not to copy configuration" was selected in step 2. Set configuration tool password, encryption password and server name. Create a Configuration Tool password to be used when configuring the 6.5 server.

Click **Next**.

6 The *User Directory Configuration* panel is displayed if LDAP User Directory mode or Windows NT User Directory mode is used. Select a **domain name style** (DNS or NetBIOS) and a **default domain**.

Note: Make sure to select an accurate **domain name style** for your system, See "External Directories and Domains" on page 77.

7 The *Summary* panel is displayed, listing the settings chosen before execution.

Click **Upgrade**.

8 The *TIBCO Spotfire Server 6.5 Upgrade* panel is displayed with a result message.

- Following a successful upgrade: "The database has been upgraded to TIBCO Spotfire Server 6.5 format" or "The database has already been upgraded to TIBCO Spotfire 6.5 format", the latter indicating that the database was already in the 6.5 format. Click **Finish**.

- If the tool encountered problems, click **Next** to see the *Upgrade Issues* panel. The issues are saved in the indicated log file. You may have to correct them manually.

Click **Finish**.

8.1.4 Run the Upgrade Tool Silently

On a server without a Graphical User Interface, you will need to run the upgrade tool silently.

1 Open the **silent.properties** file from **<6.5.0 install directory>/tools/upgrade**

2 Update **silent.properties** with the correct parameter values. There are comments throughout the file for guidance. Only the **from** parameter is mandatory.

3 Open a command prompt.

4 *Optional*: To see the parameters the upgrade tool will use:

- On Windows, type: **upgradetool.bat -h**
- On Linux, type: **upgradetool.sh -h**

Response: Response: The parameters are listed in the command prompt.

5 Run the upgrade tool silently:

- On Windows, type: **upgradetool.bat -silent   silent.properties**
- On Linux, type: **upgradetool.sh -silent   silent.properties**

6 Press **Enter**.

8.1.5 Start Spotfire Server

When the upgrade tool has completed and you have verified that there are no issues with the upgrade, start the Server. See “Start and Stop Spotfire Server” on page 110.

8.1.6 Deploy Client Packages

For Spotfire users to have their old clients automatically upgraded to version 6.5 upon next start, deploy Spotfire 6.5 packages to the server. 6.5 packages must be deployed to Spotfire Server before you can log in to it with a 6.5 client.

Refer to the *TIBCO Spotfire Deployment and Administration Manual*. If you are running Spotfire Web Player, Spotfire Automation Services, or any add-ons to the Spotfire products, also deploy these upgraded packages to the server.

8.1.6.1 Multiple Deployment Areas in Spotfire Server 6.5

Multiple Deployment Areas make it is possible to create an unrestricted number of deployment areas that can be assigned to groups. This gives the Admin fine-grained control over which users get access to which deployments. Handling of multiple deployment areas, assigning them to groups, etc. is done in the Spotfire Administration Console. See the Spotfire Administration Console Help for more information.

In a system that has been upgraded to 6.5, there will be two deployment areas named Production and Test. The Production deployment area is by default set as the default deployment area, and all existing users will get the packages deployed there.

To get the old behavior, with a production and a test deployment area and where users who previously had access to the test deployment area still get their packages from the test area, you need to deploy client packages to the test deployment area as well. Then assign this area to a group in the Spotfire system that you for instance name "test” and make sure the users who previously had access to the test area are members of the “test” group.

Note: Make sure that users who shall get the test packages only have access to the test area and are not members of the Administrator or Deployment Administrator groups since those groups have access to all areas.

If running a 5.5 or earlier client version, a user who has access to more than one deployment area gets the default deployment area.

8.1.7 Upgrade Spotfire Clients

Start a Spotfire client and log in to Spotfire Server. Make sure that the client is upgraded with the new deployment. Verify that the Spotfire library and information model are accessible and work as they did before the upgrade.

If you are running the Web Player, also upgrade the Web Player with the deployed packages. Refer to the *Spotfire Web Player Installation and Configuration Manual* for instructions on how to do this.

8.1.8 Uninstall Old Spotfire Server Version

Once you have verified Spotfire Server configuration, deployed client packages and verified the functionality of the clients, you may safely uninstall the old Spotfire Server. See “Uninstalling Spotfire Server” on page 151.

8.2 Migration from Spotfire Analytics Server 10.1

If you are running Spotfire Analytics Server 10.1, you will first need to perform a migration to Spotfire Server 3.3 to migrate information from your Spotfire Analytics Server 10.1 databases to your Spotfire Server 3.3 database. After the migration, an upgrade to Spotfire 6.5 can be performed. Please contact Spotfire Customer Support to perform the migration.

8.3 Upgrade Between Service Pack Versions

Upgrades between service pack versions, for example from 6.5.0 to 6.5.1, shall be performed by applying the latest hotfix and not by running the service pack installer. Go to **http://support.spotfire.com/patches_spotfireserver.asp** to download the latest hotfix. Installation instructions for each hotfix are included in the package. For more information, see “Hotfix Installation” on page 27.

8.4 Applying Hotfixes

Go to **http://support.spotfire.com/patches_spotfireserver.asp** to download the latest hotfix. Installation instructions for each hotfix are included in the package. For more information, see “Hotfix Installation” on page 27.

9 Backup and Restore

To enable recovery after a crash or disaster in your Spotfire system, it is important that information stored in the system is backed up. Most of this information is stored in the Spotfire database, but some of it is stored on the Spotfire Server(s) and the Spotfire Web Player server(s). This manual will not describe how to perform backups, only what to back up. It is assumed that you have some sort of backup software for files and computers, and that you use the backup tools provided with the database. Refer to the database documentation for instructions on how to perform backups.

One can only restore to a machine running the same operating system as the backed up system, since there is a bundled Java runtime with binaries for a specific architecture.

Back up each server in the cluster.

For other components in the Spotfire system, such as the Spotfire Web Player, refer to their installation manuals for instructions on how to perform backups of them.

The following sections describe what needs to be backed up.

9.1 Spotfire Database

The most important part of the Spotfire system is the Spotfire database. It contains tables which store the state of the server, for example the library, preferences, and deployments. Most of the Server configuration is also stored in the database. Even if only the database has been backed up, it is still possible to restore most of the functionality after a crash. It is therefore vital that you have a valid and current backup of the Spotfire database.

Note: Verify your backups.

9.2 Spotfire Server

A small set of configuration is unique for each Spotfire Server and it is stored on the actual Spotfire Server rather than in the database. This includes information about how Spotfire Server connects to the Spotfire database, which ports the server should listen to, authentication methods such as Kerberos etc.

During installation the server files are essentially all placed in the installation directory. It should be sufficient to back up this directory, of course it is possible to back up the entire file system.

Once a server has been configured or hotfixed there are no further persistent changes. Log files and other temporary files will change, but a restored backup will have the same functionality.

The configuration which is not in the database includes:

- Listening ports configuration (“server.xml” on page 175 for more information).

- Information about how to connect to the Spotfire database (see the section "Database Connection URL Examples" on page 181).
- Logging configuration (see the section "Log Files" on page 173).
- Memory configuration (see the section "Modifying the Virtual Memory" on page 154).
- HTTPS (see the section "HTTPS" on page 95).
- Authentication such as Kerberos or Client Certificates
- Database drivers.
- Any other advanced configuration performed in "Advanced Procedures" on page 153. When performing advanced configuration, you should always take backup into consideration.

Whenever you make any configuration changes or have applied a server hotfix, you should also perform a backup of the Spotfire Server installation directory.

Windows Installations

On Windows installations, there is functionality which will not be restored by only recovering the server installation directory:

- Windows Service
- Uninstall functionality
- Start Menu shortcuts

The Windows Service can be (re-)installed using the bat file **service.bat** located in the **<installation dir>\tomcat\bin** directory. Run it from a command line with the following arguments: **C:\tibco\tss\6.5\tomcat\bin>service.bat install**

Uninstallation can be done by removing the service and simply remove the installation directory.

The Start Menu shortcuts can be backed up by copying them to the server installation directory, back that up, and when restoring, copying these files to the start menu directory.

Unix and Linux Installations

On Unix and Linux installations, no essential data is placed outside the installation directory by Spotfire Server. If you have a startup scripts for the server, it will need to be recreated.

Network Considerations

If you are using Kerberos you should note that configuration needed for this to work is tied to a specific machine and cannot be copied easily to a new one.

You should also consider any other conditions in your environment and their implications, such as IP addresses and firewall rules, LDAP restrictions, and anything else that might affect getting a system back up and running.

9.3 Disaster Recovery

In the event of a disaster, where you have no valid backups available, you should think of the following:

- Without a valid backup of the Spotfire database, you will not be able to restore your Spotfire system to its previous state. You must therefore have a valid and current backup of your Spotfire database.
- Without a valid backup of the Spotfire Server(s), you can get a functional environment by installing a new Spotfire Server, make sure that it can connect to the database, and configure the things which are stored locally.

10 Uninstallation

A complete uninstallation requires two steps: Uninstalling Spotfire Server and removing the database. The procedures for Windows, Linux and Solaris, and the actions for SQL Server and Oracle respectively are described in this chapter.

If you have placed any additional files in the installation directory or any of its subfolders, such as Spotfire Library export files, you should move these files to a secure location before uninstalling. The installer will remove the installation folder and all its subfolders.

10.1 Uninstalling Spotfire Server

10.1.1 Windows

Interactive Uninstallation

Uninstallation is performed through the regular Windows procedure selecting: **Start > Control Panel > Programs and Features > Uninstall or change a program** and then right-clicking **Spotfire Server 6.5** and selecting **Uninstall**.

After successful uninstallation, only user modified files remain on the machine (such as custom JDBC drivers).

Silent Uninstallation

Run the MSI mechanism as follows:

```
msiexec.exe  /qn  /x  {A5D6CFD2-31FA-4A3E-8D4C-8F5486FE4F6D}
```

10.1.2 RPM Linux

Uninstallation is performed via the command: **rpm -e tss-6.5**

After a successful uninstallation, only modified files in **tomcat/conf** remain.

10.1.3 Tarball Linux

If the server was configured to start on boot, it must be stopped and removed.

To stop the server, run the command:

```
service tss-6.5 stop
```

To remove the server, run the command:

```
chkconfig --del tss-6.5
```

Delete added scripts by running the following commands:

```
rm /etc/init.d/tss-6.5
```

rm /etc/sysconfig/tss-6.5

Note: To be able to do this, the Spotfire Administrator must have **root** access.

The next step is to remove the folder with Spotfire Server files. Do this by running the command:
rm -rf <the folder where the tarball was installed>

10.1.4 Solaris

An uninstallation application is created when installing: **<installation dir>/installer/uninstaller/Uninstall_TIBCO_Spotfire_Server_6.5**. Run the application to uninstall.

Remove the startup script **/etc/init.d/tss** manually if you have installed it. Also make sure to remove the symbolic links in the appropriate runlevel directories (for example **/etc/rc2.d**).

10.2 Removing the Database

In the **scripts/oracle_install/utilities** and **scripts/mssql_install/utilities** folders on Spotfire Server installation kit, there are a number of scripts that can be used to remove the Spotfire and Demo databases. Before you run the script, open it in a text editor and edit the variables set during database preparation. See the section “Prepare the Database” on page 17 for detailed information about these variables.

WARNING

Removing the database deletes all user data and most Spotfire Server configurations permanently.

Microsoft SQL Server

- **drop_databases.bat**: If you are using database authentication with your Microsoft SQL server.
- **drop_databases_ia.bat**: If you are using Windows Integrated Authentication with your Microsoft SQL Server.

Oracle

- **drop.databases.bat**: If you are running Oracle on a Windows server.
- **drop_databases.sh**: If you are running Oracle on a Unix server.

11 Advanced Procedures

This section describes advanced manual procedures for setting up various features supported by the Spotfire system. Many of the procedures assume prior knowledge about LDAP directory, Kerberos, Windows Server, Apache httpd, etc. For detailed information about how these various technologies work and how they are set up, refer to the documentation for the specific technology.

11.1 Running Database Preparation Scripts Manually

The automatic **create_databases** script requires that your database engine supports user name and password authentication. If your database for some reason does not support this, because, say, you are using Kerberos authentication, you must run the SQL preparation scripts manually.

The scripts to run:

- **create_server_db.sql**
- **populate_server_db.sql**
- **create_server_user.sql**

Note: Demo SQL scripts are only necessary if you want to install the demo database tables shipped with Spotfire Server.

- **create_demotables.sql**
- **create_demo_user.sql**
- All the SQL files in the folder demodata

Read through the **create_databases** script to understand how they work. Below are some notes regarding the supported database server types.

Oracle

When populating a Spotfire database on an Oracle Server the **create_databases** script passes certain variables to these scripts. These variables include:

- ROOTFOLDER
- CONNECTIDENTIFIER
- SERVER_DATA_TABLESPACE
- SERVER_TEMP_TABLESPACE

When you run the SQL scripts manually, you must make sure to pass these variables along to the scripts.

Microsoft SQL Server

When populating a Spotfire database residing on a Microsoft SQL Server the create_databases script passes certain variables to these scripts. These variables include:

- SERVERDB_NAME
- DEMODB_NAME

When you run the SQL scripts manually, you must make sure to pass these variables along to the scripts.

11.2 Resizing Temporary Tablespace

The tablespaces/database files for Spotfire Server using Oracle/MSSQL Database uses autoextend/autogrowth by default. If this is inappropriate for your needs alter these settings. It might be desirable to alter the amount the files should be extended by with each increment. For Oracle there is a maxsize for each tablespace which should be reviewed. For MSSQL, there is an unlimited growth this should also be reviewed.

11.3 Modifying the Virtual Memory

If many simultaneous users intend to perform heavy data pivoting via Information Services or in other ways stress the server, you may need to modify the amount of memory available to the virtual machine.

▶ To set up the start script when not running as a Windows service:

1. Open the file **<installation dir>/tomcat/bin/setenv.bat/.sh** in a text editor.
2. Locate the line that sets the variable JAVA_OPTS:

```
set JAVA_OPTS=-server -XX:+DisableExplicitGC -XX:MaxPermSize=256M -Xms512M
-Xmx1536M
```

   or

```
JAVA_OPTS="-server -XX:+DisableExplicitGC -XX:MaxPermSize=256M -Xms512M
-Xmx1536M"
```

3. Alter the -xms and the -Xmx values **-Xms512M -Xmx1536M** to the amount of memory you wish to allocate.
4. Restart the server.

▶ To set up the start script when running as a Windows service:

1. Stop the Spotfire Server service
2. Go to the **<installation dir>/tomcat/bin directory**
3. Run the command: **service.bat remove**
4. Edit the **<installation dir>/tomcat/bin/service.bat** file.

5 Look for the entries: **--JvmMs 512 --JvmMx 1536**

6 Alter **512** and/or **1536** to suitable memory values (in MB).

7 Run the command: **service.bat install**

8 Start the Spotfire Server service.

11.4 Storing Library Content Outside of the Spotfire Database

To minimize the size of your TIBCO Spotfire database, you can store your organization's Spotfire library content (analyses and analysis data) in the cloud using Amazon Web Services S3 (AWS), or in a file system elsewhere.

In a typical Spotfire installation, the largest part of database storage consists of the library content. When you move the library content to external storage, only the metadata about the library files remains in the database. The other items in database storage—system configuration data, permissions, licenses, and so on—remain where they are.

Note: In this scenario, *all* library content is stored externally; it isn't possible to split storage between the server database and the external site.

Currently there are three main drawbacks to this option:

- Referential integrity is not guaranteed; there is the possibility that content referenced in the Spotfire database will not exist in external storage, and vice versa.
- Your system may run more slowly, such as when loading files.
- A database backup will not back up the library content.

11.4.1 Configuring External Library Storage

External library storage is configured using the command line tool.

Prerequisite:

- To use AWS, you need an Amazon S3 account.

1 Back up the database.

2 Export the library. See “export-library-content” on page 247 for more information.

3 Remove the content in the library.
Note: Do not use the **truncate** command in the database because there are hidden folders that should not be removed.

4 Configure Spotfire Server to use external library storage:

a) To enable external storage and select the type of external storage, use the command **config-library-external-data-storage**. For more information about this command, see “config-library-external-data-storage” on page 217.

b) To configre AWS storage, use the command **config-library-external-s3-storage**. To use this command, you must have:

- Information about the Amazon S3 account that you will use.
- A bucket name. Every server database (or database cluster) should have its own bucket. (Items stored in S3 are identified by their GUIDs. If different servers use the same bucket, importing files to Cluster B—when the files already exist in Cluster A—will overwrite the files in Cluster A.)

You can set the following options when using this command:

- Which AWS regional datacenter the server should connect to.
- Whether large files should be uploaded in chunks, and the details of this behavior.

For details about this command, see “config-library-external-s3-storage” on page 219.

-OR-

b) To configure external library storage in a file system, use the command **config-library-external-file-storage** to specify the path to the storage root. Subdirectories for the content files are created under this root.

For details about this command, see “config-library-external-file-storage” on page 219.

5 Import the library.

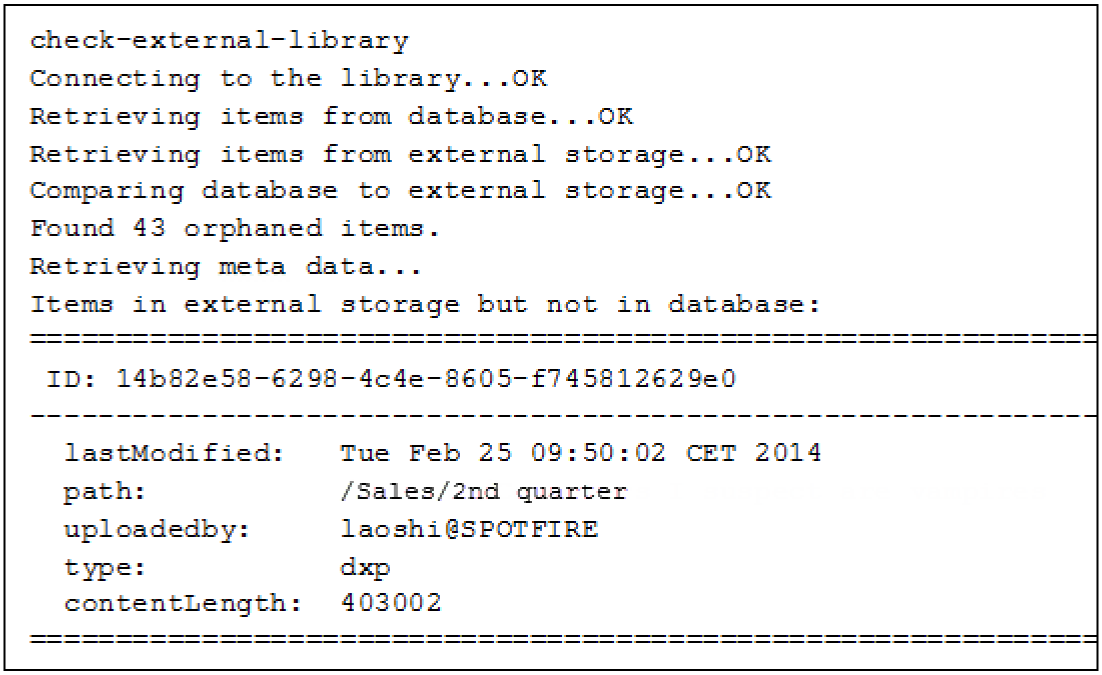
Note: Both external library storage systems use the Spotfire library globally unique identifiers (GUIDs) to identify files.

11.4.2 Monitoring External Library Storage and Fixing Inconsistencies

Because there is no guarantee of referential integrity when using external library storage, the administrator should regularly check for inconsistencies between the metadata in the TIBCO Spotfire database and the files in external storage.

External library storage is monitored using the command line tool.

1 Use the command **check-external-library** to check for discrepancies. A discrepancy report is generated, including where discrepancies occur and any available information to help identify the "orphan" files. This is an excerpt from a report:



```
check-external-library
Connecting to the library...OK
Retrieving items from database...OK
Retrieving items from external storage...OK
Comparing database to external storage...OK
Found 43 orphaned items.
Retrieving meta data...
Items in external storage but not in database:
==============================================================
 ID: 14b82e58-6298-4c4e-8605-f745812629e0
--------------------------------------------------------------
  lastModified:   Tue Feb 25 09:50:02 CET 2014
  path:           /Sales/2nd quarter
  uploadedby:     laoshi@SPOTFIRE
  type:           dxp
  contentLength:  403002
==============================================================
```

For details on this command, see “check-external-library” on page 197.

2 If a file is found in external storage that is not referenced in the Spotfire database, you can download the file. If it is an analysis file, you can then manually save it to the Spotfire library. If metadata is found for a file that does not exist, you can delete the metadata.

If you want to	Do this	For more information, see
Retrieve an orphan file from Amazon Web Services S3 (AWS)	Download it using the command **s3-download**.	“s3-download” on page 273.
Retrieve an orphan file from an external file system	Manually copy it from the file system.	
Delete files from the AWS	Use the command **delete-library-content**.	“delete-library-content” on page 241.
Delete files from an external file system	Manually delete the files.	
Delete metadata from Spotfire Server	Use the command **delete-library-content**.	“delete-library-content” on page 241.

11.5 Data Source Templates

One feature of the Spotfire system is the ability to create “information links“, easy-to-use aids for users to access data stored in databases. Using the Information Designer tool found in the Spotfire client, a database administrator can configure which data sources should be available to choose from when creating such information links.

When a connection is made to a data source it needs information about which type of database is used, as well as which type of database driver is used. To make it easier to set up multiple data sources in Information Designer, there are a set of “data source templates” with predefined settings that can be used later in Information Designer. You can also create custom data source templates. All data source templates are specified using the configuration tool.

A data source template is an XML configuration. This XML configuration includes a number of settings that customize the way information links interact with the data source. Read more about the XML format in “XML Settings” on page 159.

The following data source templates are available by default:

- **Oracle (DataDirect driver)**
- **Microsoft SQL Server (DataDirect driver)**

When adding more data source templates, they can be based on the following types:

Teradata
Sybase (JTDS)
Sybase (DataDirect)
Sybase
SQL Server 2005
SQL Server (JTDS)
SQL Server (DataDirect)
SQL Server
SAS/SHARE
Composite
Oracle (delegated Kerberos)
Oracle (DataDirect)
Oracle
MySQL5
MySQL (DataDirect)
MySQL
DB2 (DataDirect)
DB2

Note: For the MySQL5 vendor driver to work with MySQL data sources that includes TIMESTAMPS that can potentially be null, you have to edit the template to get it to work:

Locate the following section:

```
<connection-properties>
  <connection-property>
    <key>useDynamicCharsetInfo</key>
    <value>false</value>
  </connection-property>
</connection-properties>
```

and add the following within the **connection-properties** tag:

```
<connection-property>
  <key>noDatetimeStringSync</key>
  <value>true</value>
</connection-property>
```

```
<connection-property>
  <key>zeroDateTimeBehavior</key
  <value>convertToNull</value>
</connection-property>
```

This is due to a restriction in MySQL5 that has to do with null values in TIMESTAMPS.

11.5.1 Handling Data Source Templates

If you add a data source template that does not use the pre-installed DataDirect driver, you must manually install this driver on each Spotfire Server in the cluster before you restart the cluster. Download the appropriate driver JAR file and place it in the **<installation dir>/tomcat/lib** folder of each server.

▸ To add a new data source template:

Use the command "add-ds-template" on page 193.

▸ To enable, modify or disable a data source template:

For a data source template to become available in the Information Designer, it must be enabled. Use the command "modify-ds-template" on page 268.

▸ To remove a data source template:

Verify that no data sources use the data source template before you remove it. If a data source template is removed, all data sources using that template will stop working.

Use the command "remove-ds-template" on page 270.

11.5.2 XML Settings

The table below shows all settings available. Only the first three are mandatory:

- type-name
- driver
- connection-url-pattern

If left out, all other settings will automatically use their default values.

Setting	Description
type-name	A unique name for the configuration.
driver	The JDBC driver Java class used for creating connections.
connection-url-pattern	A pattern for the connection URL. The URL syntax is driver specific.
ping-command	A dummy command to test connections. Default: **SELECT 1**
connection-properties	JDBC connection properties.

metadata-provider
Java class that provides database metadata.
See Spotfire Technology Network.
Default: **BasicJDBCMetadataProvider**

sql-filter
Java class that generates SQL.
See Spotfire Technology Network.
Default: **BasicSQLFilter**

sql-runtime
Java class that handles SQL execution.
See Spotfire Technology Network.
Default: **BasicSQLRuntime**

fetch-size
A fetch size specifies the amount of data fetched with each database round trip for a query. The specified value is shown as the default value in Information Designer. May be changed at instance level.
Default: **10000**

batch-size
A batch size specifies the amount of data in each batch update. The specified value is shown as the default value in Information Designer. May be changed at instance level.
Default: **100**

max-column-name-length
The maximum length of a database column name. This limit is used when creating temporary tables.
Default: **30**

table-types
Specify which table types to retrieve.
Default: **TABLE, VIEW**

supports-catalogs
Tells if the driver supports catalogs.
Default: **true**

supports-schemas
Tells if the driver supports schemas.
Default: **true**

supports-procedures
Tells if the driver supports stored procedures.
Default: **false**

supports-distinct
Tells if the driver supports distinct option in SQL queries.
Default: **true**

supports-order-by
Tells if the driver supports order-by option in SQL queries.
Default: **true**

column-name-pattern
Determines how a column name is written in the SQL query.
Default: **“$$name$$”**

table-name-pattern
Determines how a table name is written in the SQL query.
Default: **“$$name$$”**

schema-name-pattern
Determines how a schema name is written in the SQL query.
Default: **“$$name$$”**

catalog-name-pattern Determines how a catalog name is written in the SQL query.
Default: **“$$name$$”**

procedure-name-pattern Determines how a procedure name is written in the SQL query.
Default: **“$$name$$”**

column-alias-pattern Determines how a column alias is written in the SQL query.
Default: **“$$name$$”**

string-literal-quote The character used as quote for string literals; SQL-92 standard.

max-in-clause-size The maximum size of an SQL IN-clause. Larger lists are split into several clauses that are OR:ed together.
Default: **1000**

condition-list-threshold A temporary table is used when executing an SQL query, where total size of a condition list is larger than this threshold value. A Data Base Administrator may prefer a lower value than the default. Depends on the maximum SQL query size.
Default: **10000**

expand-in-clause If true, an SQL **IN**-clause will be expanded into **OR** conditions.
Default: **false**

table-expression-pattern Determines how a table expression is written in the SQL query; **catalog** and **schema** may be optional (surrounded by brackets).
Default: **[$$catalog$$.][$$schema$$.]$$table$$**

procedure-expression-pattern Determines how a procedure expression is written in the SQL query.
Default: **[$$catalog$$.][$$schema$$.]$$procedure$$**

procedure-table-jdbc-type Integer representing the jdbc type identifying a table returned form a procedure as defined by **java.sql.Types**.
Default: **0**

procedure-table-type-name Display name for tables from procedure. This is currently not visible to the user in any UI.
Default: **null**

date-format-expression An expression that converts a date field to a string value on the format: **YYYY-MM-DD**, for example, **2002-11-19**. Used in **WHERE** and **HAVING** clauses. The tag **$$value$$** is a placeholder for the date field.
Default: **$$value$$**

date-literal-format-expression An expression that converts a date literal on the format **YYYY-MM-DD** to a date field value. Used in **WHERE** and **HAVING** clauses. The tag **$$value$$** is a placeholder for the date literal.
Default: **'$$value$$'**

time-format-expression

An expression that converts a time field to a string value on the format: **HH:MM:SS**, for example **14:59:00**. Used in **WHERE** and **HAVING** clauses. The tag **$$value$$** is a placeholder for the time field.
Default: **$$value$$**

time-literal-format-expression

An expression that converts a time literal on the format **HH:MM:SS** to a time field value. Used in **WHERE** and **HAVING** clauses. The tag **$$value$$** is a placeholder for the time literal.
Default: **'$$value$$'**

date-time-format-expression

An expression that converts a datetime field to string value on the format: **YYYY-MM-DD HH:MM:SS**, e.g **2002-11-19 14:59:00**. Used in **WHERE** and **HAVING** clauses. The tag **$$value$$** is a placeholder for the date-time field.
Default: **$$value$$**

date-time-literal-format-expression

An expression that converts a date-time literal on the format **YYYY-MM-DD HH:MM:SS** to a date-time field value. Used in **WHERE** and **HAVING** clauses. The tag **$$value$$** is a placeholder for the date-time literal.
Default: **'$$value$$'**

java-to-sql-type-conversions:

String
Integer
Long
Float
Double
Date
Time
DateTime

Type conversions needed when a join data source creates a temporary table for result from a subquery. For String conversion **%s** will be replaced by the size of the string. A match-length attribute may be specified (see MySQL). Different String types may be needed dependant of the length of the string. Note that there must be a **VARCHAR** conversion for when the length of the string is unknown (255 in the example here). When several **VARCHAR** mappings are specified, the mapping that first matches the match-length is used.
Default: **VARCHAR($$value$$) VARCHAR(255) INTEGER BIGINT REAL DOUBLE PRECISION DATE TIME TIMESTAMP**

temp-table-name-pattern

Determines how to format a temporary table name in an SQL command.

Default: **$$name$$**

create-temp-table-command

SQL commands for creating a temporary table. This is used to store filter values (when more than **condition-list-threshold**) and to store result from subqueries. The syntax may vary between databases. **$$name$$** is a placeholder for the table name. **$$column_list$$** is a placeholder for a column list on the format **(name type, name type, ...)**.
Default: **CREATE TEMPORARY TABLE $$name$$ $$column_list$$**

drop-temp-table-command

SQL commands for deleting a temporary table. The syntax may vary between databases. **$$name$$** is a placeholder for the table name.
Default: **DROP TABLE $$name$$**

data-source-authentication
Default value data source authentication. (boolean). This value can be set (overridden) in the Information Interaction Designer.
Default: **false**

lob-threshold
Threshold when LOB values used as parameters in a WHERE clause, must be written in temporary tables. The default means no limit.
Default: **-1**

use-ansii-style-outer-join
The default generated SQL uses the Oracle way with “(+)” to indicate joins. If this setting is set to true an attempt is made to rewrite it to standard ANSII format, making it possible to run on non Oracle databases
Default: **false**

credentials-timeout
Defines the time in seconds user credentials are cached on the server for a particular data source. Value must be between 900 (15 minutes) and 604800 (1 week). Applicable only if **data-source-authentication** is set to **true**.

Default:**86400** (24 hours)

11.5.2.1 Defining JDBC Connection Properties

The optional <connection-properties> parameter block in the <jdbc-type-settings> configuration can be used to define JDBC connection properties parameters to be used when connecting to the data sources of the given type. A typical use case is to specify encryption and integrity checksum algorithms for secure database connections.

Each connection property consists of a key-value pair. The syntax for specifying JDBC connection properties for a connection pool is shown in the configuration example below.

If you need different JDBC connection properties for different data sources of the same type, just duplicate the <jdbc-type-setting> configuration, rename the configurations for each variant needed and define the proper JDBC connection properties. Make sure to update any already existing data sources so that they are of the correct type.

Example: Defining JDBC Connection Properties for data source of type **oracle**. This example creates an encrypted connection to the database.

```
<jdbc-type-settings>
     <type-name>oracle</type-name>
     <driver>oracle.jdbc.OracleDriver</driver>
     <connection-url-
pattern>jdbc:oracle:thin:@&lt;host&gt;:&lt;port1521&gt;:&lt;sid&gt;</
connection-url-pattern>
     <ping-command>SELECT 1 FROM DUAL</ping-command>
     <connection-properties>
        <connection-property>
          <key>oracle.net.encryption_client</key>
          <value>REQUIRED</value>
        </connection-property>
```

```
          <connection-property>
            <key>oracle.net.encryption_types_client</key>
            <value>( 3DES168 )</value>
          </connection-property>
          <connection-property>
            <key>oracle.net.crypto_checksum_client</key>
            <value>REQUIRED</value>
          </connection-property>
          <connection-property>
            <key>oracle.net.crypto_checksum_types_client</key>
            <value>( MD5 )</value>
          </connection-property>
       </connection-properties>
        ...
</jdbc-type-settings>
```

11.5.2.2 Advanced Connection Pool Configuration

Information Services uses the same underlying connection pool implementation as Spotfire Server uses for connecting to its own database. The following special parameters are available to configure some of the aspects of that connection pool:

- **spotfire.pooling.data.source.scheme**
  Corresponds to the **pooling-scheme** parameter. See“Database Connectivity” on page 179”.
- **spotfire.pooling.data.source.connection.timeout**
  Corresponds to the **connection-timeout** parameter. See“Database Connectivity” on page 179”.
- **spotfire.pooling.data.source.login.timeout**
  Corresponds to the **login-timeout** parameter. See“Database Connectivity” on page 179”.
- **spotfire.kerberos.login.context**
  Corresponds to the **kerberos-login-context** parameter.

All these parameters should be added as JDBC connection properties. However, they will never be used as real JDBC connection properties and will never be sent to a database server.

Example: Configuring a connection pool for Oracle databases

```
 <jdbc-type-settings>
      <type-name>oracle</type-name>
      <driver>oracle.jdbc.OracleDriver</driver>
      <connection-url-
pattern>jdbc:oracle:thin:@&lt;host&gt;:&lt;port1521&gt;:&lt;sid&gt;</
connection-url-pattern>
      <ping-command>SELECT 1 FROM DUAL</ping-command>
      <connection-properties>
        <connection-property>
          <key>spotfire.pooling.data.source.scheme</key>
          <value>WAIT</value>
        </connection-property>
        <connection-property>
```

```
          <key>spotfire.pooling.data.source.connection.timeout</key>
          <value>1800</value>
        </connection-property>
        <connection-property>
          <key>spotfire.pooling.data.source.login.timeout</key>
          <value>30</value>
        </connection-property>
      </connection-properties>
      ...
</jdbc-type-settings>
```

11.5.2.3 Using Kerberos Authentication for JDBC Data Sources

Configuration of Kerberos authentication for JDBC data source is performed in a similar way as for the connection to the Spotfire database. See section “Using Kerberos to Log In to the Spotfire Database” on page 100 for more information.

Example: Configuring a connection pool for Oracle databases

```
<jdbc-type-settings>
      <type-name>oracle</type-name>
      <driver>oracle.jdbc.OracleDriver</driver>
      <connection-url-
pattern>jdbc:oracle:thin:@&lt;host&gt;:&lt;port1521&gt;:&lt;sid&gt;</
connection-url-pattern>
      <ping-command>SELECT 1 FROM DUAL</ping-command>
      <connection-properties>
        <connection-property>
          <key>spotfire.kerberos.login.context</key>
          <value>DatabaseKerberos</value>
        </connection-property>
        <connection-property>
          <key>oracle.net.authentication_services</key>
          <value>( KERBEROS5 )</value>
        </connection-property>
      </connection-properties>
      ...
</jdbc-type-settings>
```

11.5.2.4 Using Kerberos Authentication with Delegated Credentials

To make users authenticate to different data sources with their own single sign-on login information, the server can delegate the user authentication to the data source. This is only possible if you use the Kerberos single sign-on method.

In order to set this up, there are a number of steps that must be taken. Each of these steps is described in detail below.

1 Set up Kerberos authentication as described in the section “Kerberos Authentication” on page 66. Make sure that users are able to log in with this method.

2 Grant the Spotfire Server service account used for client authentication the right to delegate client credentials.

3 Create a JDBC data source template using Kerberos login

Grant the Spotfire Server Service Account the Right to Delegate Client Credentials

If your Window Domain is using Windows Server 2003 or later, grant constrained delegation rights to the service account: Only the specified accounts can be delegated by the service account. If you are using an earlier version of Windows Server or can't use this method, grant unconstrained delegation rights. Both methods are described below.

Note: In order for delegation to work, you must also ensure that no client user account in the domain has the setting **Account is sensitive and cannot be delegated**. By default, this is not set.

▸ To enable constrained delegation:

1 On the Domain Controller, select **Start > Programs > Administrative Tools**.

2 Select **Active Directory Users and Computers**.

3 Locate the account.

4 Right-click the account name, and then click **Properties** to open the account properties.

5 On the **Delegation** tab, select **Trust this user for delegation to specified services only**.
Note: The Delegation tab is only visible for accounts that SPNs are mapped to.

6 Select **Use any authentication protocol**.

7 Click **Add**.

8 Click **Users or Computers** and select the account that Spotfire Server has a keytab for and the SPNs are mapped to.

9 Select all services that apply and click **OK**.

10 Click **Apply**.

▸ To enable unconstrained delegation for a on a Domain Controller in Windows 2000 Mixed or Native Mode:

1 On the Domain Controller, select **Start > Programs > Administrative Tools**.

2 Select **Active Directory Users and Computers**.

3 Locate the account.

4 Right-click the account name, and then click **Properties** to open the account properties.

5 Select the **Account** tab and select **Account is trusted for delegation** in the **Account Options** list.

6 Click **Apply**.

▶ To Enable Unconstrained Delegation on a Domain Controller in Windows Server 2003 Mode:

1 On the Domain Controller, select **Start > Programs > Administrative Tools**.

2 Select **Active Directory Users and Computers**.

3 Locate the account.

4 Right-click the account name, and then click **Properties** to open the account properties.

5 On the **Delegation** tab, select **Trust this user for delegation to any service (Kerberos only)**, and then click **Apply**.
Note: The Delegation tab is only visible for accounts that SPNs are mapped to.

Create an Information Services data source template using Kerberos login

The default Information Services Data Source templates shipped with Spotfire Server are not configured to use Kerberos. You must therefore create a new data source template based on one shipped.

▶ To Create an Information Services data source using Kerberos login:

1 Use the **list-ds-template** (page 257) command to list the existing data source templates and select one that matches the database you are setting up, for example Oracle.

2 Use the **export-ds-template** (page 245) command to export the definition of the selected data source template.

3 In a text editor, open the exported definition file.

Add the JDBC connection property key **spotfire.connection.pool.factory.data.source** with the value **kerberos.data.source** within the **connection-properties** element. If there is no **connection-properties** element, create one.

There may also be other connection properties you must add - consult the documentation of the database server for more information. See “Defining JDBC Connection Properties” on page 163 for general instructions about adding connection properties.

Example:

```
<jdbc-type-settings>
  <type-name>oracle</type-name>
  <driver>oracle.jdbc.OracleDriver</driver>
  <connection-url-
pattern>jdbc:oracle:thin:@&lt;host&gt;:&lt;port1521&gt;:&lt;sid&gt;</
connection-url-pattern>
  <ping-command>SELECT 1 FROM DUAL</ping-command>
<connection-properties>
  <connection-property>
    <key>spotfire.connection.pool.factory.data.source</key>
    <value>kerberos.data.source</value>
  </connection-property>
  <connection-property>
    <key>oracle.net.authentication_services</key>
    <value>(KERBEROS5)</value>
  </connection-property>
</connection-properties>
```

4 Use the **add-ds-template** command (page 193) to add the new data source template with a suitable name, like **oracle_kerberos**, using the modified template definition.

5 Import the configuration and restart the server.

11.5.2.5 Verifying a Data Source Template

To verify the data source template from Spotfire:

1 Log in to Spotfire as an administrator.

2 Select **Tools > Create Information Link**.

3 Click on the **Setup Data Source** link.

4 Enter a name for the data source connection.

5 Specify the type of data source.

6 Enter the **connection URL**.

7 Enter **max/min-values** for the connection pool.

8 Enter a **username** and a **password** to connect to the database.
Note: Does not apply to Kerberos.

9 Click **Save**.

10 Click on the **Data sources** tab in the left pane.

Response: The data source name should appear in the tree to the left, ready for use.

11.6 Information Services Settings

Information Services provides end users with the ability to access and pivot data from multiple databases simultaneously, without having to know anything about installing database drivers, underlying data schemas or SQL.

End users' access to data from multiple sources can be configured and controlled through settings in Information Services. Below is a list of common settings and short descriptions. See “To edit the configuration.xml file” on page 32 on how to change the settings.

Setting	Description
information-services.jdbc.oracle.use-faster-schema-listing	List all Oracle users as schema list.

information-services.dat.no-sbdf	Use spotfire text data format or spotfire binary data format when transferring data from Spotfire Server to a Spotfire client.
information-services.runtime-query-validation	Validate information link prior to execution.
information-services.dat.data-block-queue-size	Maximum number of queued (not yet consumed by client) data blocks per job
information-services.dat.idle-limit	Maximum idle time in seconds before a job is garbage collected.
information-services.dat.max-field-size	Maximum size (in Megabytes) for a data cell.
information-services.dat.max-jobs	Maximum number of concurrent jobs.
information-services.dat.max-timeout	Maximum value of timeout parameters; must be at least 60 seconds more than the idle limit.
information-services.dat.pivot.thread-pool-size	Maximum number of pivot worker threads
information-services.dat.reshape.max-memory-usage	Maximum memory available to a reshape operation.
information-services.dat.retrieve-timeout	Maximum time allowed for retrieve requests, in seconds.
information-services.dat.thread-pool-size	Maximum number of job worker threads.
information-services.ds.credentials-cache-timeout	The default expiration time in seconds for cached data source authentication credentials.
information-services.ds.credentials-provider	The class used to provide credentials for datasources that require authentication.
information-services.jdbc.connection-login-timeout	Login timeout for JDBC database connections.

information-services.jdbc.oracle.temp-table-grantee	Selecting priviliges on temporary tables used during query execution will be granted to this user or role. The temporary tables are only valid during the query transaction.
information-services.jdbc.use-inner-select-in-clause	This setting affects the behaviour when the number of filter values sent to a jdbc data source exceeds the condition-list-threshold. If set to false (default): all data rows matching any duplicate filter values will be duplicated, If set to true: data rows matching any duplicates will not be duplicated (the same behaviour as when the number of filter values is below the condition-list-threshold limit), but there is a large performance penalty.

Example: How to enable the **"information-services.jdbc.oracle.use-faster-schema-listing"** setting:

1 Open the configuration.xml file in a text editor and locate the "open <oracle>" tag.

2 Add "**<use-faster-schema-listing>true</use-faster-schema-listing>**". See below:

```
<configuration ...>
...
  <information-services>
  ...
    <jdbc>
    ...
      <oracle>
      ...
        <use-faster-schema-listing>true</use-faster-schema-listing>
        ...
      </oracle>
      ...
    </jdbc>
    ...
  </information-services>
  ...
</configuration>
```

3 Save the **configuration.xml** file.

4 Import the **configuration.xml** file.

5 Restart Spotfire Server.

Note: All Oracle users will be listed as a schema list, even if they are not editable.

11.7 Default Join Database

The default join database is used for creating temporary tables and joining the final result when running an information link. Most often using the standard Spotfire database for the default join database will work fine. However, in certain situations you may want to configure another database to be used. For example, if you prefer to run these operations as a specific user on the database, or if you want to use a database that is specifically optimized for temporary tables.

To set up a default join database use the command “create-join-db” on page 230.

Default Join Database Settings

Option	Description
Type	Sets the type of database and driver you want to use as the default join database. Refers to a data source template.
Connection URL	The connection URL to the database.
Number of Connections	A minimum and maximum number of connections to use when accessing the database.
Username and Password	The User name and Password that will be used to access the database.

11.8 Spotfire Server Public Web Service API's

It is possible to build specific functionality that can call Spotfire Server through a set of public Web Service API's. These can be accessed at:

- **http[s]://<tss_host>[:<port>]/spotfire/ws/pub/LibraryService**
- **http[s]://<tss_host>[:<port>]/spotfire/ws/pub/SecurityService**
- **http[s]://<tss_host>[:<port>]/spotfire/ws/pub/UserDirectoryService**

A description of each web service (a WSDL file) can be retrieved by appending **?wsdl** to each web service URL. The WSDL files can be used to generate client proxies which will contain all types and methods that may be used. The implementing classes may not be called directly from Java code.

11.8.1 Enable the API

Before the API can be used it must be enabled by setting the **public-api.web-services.enabled** configuration property to **true**. All users calling the API must also be members of the API User group.

11.8.2 Generate Client Proxies

Proxies can be generated using a tool of your choice. Here is an example on how to do it using the wsimport tool that is included with the Oracle JDK 7.

1. Create an authentication file containing the URL of each web service, including a valid user name and password of a user that is a member of the API User group.
2. Generate the proxies by running wsimport for each web service (specifying the authentication file created in the previous step).

Examples of authentication files:

- **http://user:password@tss.example.com:8080/spotfire/ws/pub/LibraryService?wsdl**
- **http://user:password@tss.example.com:8080/spotfire/ws/pub/SecurityService?wsdl**
- **http://user:password@tss.example.com:8080/spotfire/ws/pub/UserDirectoryService?wsdl**

Examples on how to generate the proxies, using the authentication files above:

- **wsimport -d bin -s src -Xauthfile auth.txt http://tss.example.com:8080/spotfire/ws/pub/LibraryService?wsdl**
- **wsimport -d bin -s src -Xauthfile auth.txt http://tss.example.com:8080/spotfire/ws/pub/SecurityService?wsdl**
- **wsimport -d bin -s src -Xauthfile auth.txt http://tss.example.com:8080/spotfire/ws/pub/UserDirectoryService?wsdl**

Reference: Spotfire Server Files

This reference describes a select set of files on Spotfire Server. Some are accessed for information, other for configuration.

SF.1 Server Logs and Diagnostics

The main purpose of logging is to aid in the detection, diagnosis and resolution of any problems the server experiences. Therefore, in the normal operation of the server, a minimal amount of logging is enabled.

For each Spotfire Server you can view logs and diagnostics in the Server Logs and Diagnostics tool, available at the URL **http://<server>/spotfire:<port>**. Log in with a user that has either Spotfire Administrator or Spotfire Diagnostics Administrator permissions. For more information about Spotfire permissions and administrator roles, see the *TIBCO Spotfire Deployment and Administration Manual*.

- Server Logs provide information about what is happening on the server, such as access logs, SOAP communication, Information Links that are used, and so forth.
- Diagnostics provide information about the server itself, such as information about the Database Server, Operating System, Spotfire Server, Application Server, System, and Java version.

Log Files

Spotfire Server uses rolling logs, which means that when a log file gets too big it splits into several files. These are indexed by a number, (the higher the number, the older the log) and can be selected in the drop-down list in the Server Log Files interface. When a rolled log file reaches a certain number it is deleted.

The log files are located in the **<installation dir>/tomcat/logs** directories. There are several log files that you can configure and view:

Log Name	File Name	Contents
Configuration Tool Log	**tools.log**	Information about the activity of the configuration tool.
Information Services Usage Log	**isusage.log**	Information about Information Services usage.
Library Log	**library.log**	Information about Spotfire Library usage.
Library Import/ Export Log	**impex.log**	Information about Spotfire Library imports and exports.
SOAP Log	**soap.log**	Information about SOAP communication.
SQL Log	**sql.log**	Information about SQL expressions performed when an information link is executed.

Server Log	**server.log**	Information about all activity on the server except those events recorded in the Server Access Log.
Server Access Log	**access.log**	Information about client access and access attempts to the server and files in the library.
Server Diagnostics Log	**server-diagnostics.log**	Diagnostic information about server measures.
Server Usage Log	**usage.log**	Information about client access and access attempts to the server.
Startup Log	**startup.log**	Information about JAR files loaded on server startup.

▸ Viewing Logs:

To view a log, log in to the Spotfire Administration Console and click **Open Logs & Diagnostics**, select it from the drop-down list below the text **View log file**. If the log is very long, it will be paginated. Use the links located above and below the log display to view the entire log.

Log Configuration Files

You can determine what should be logged in the log files by applying settings in a certain Log Configuration File. This configuration file will set the level of detail for the actual log files. To do this, first select the Log File in the drop-down menu below Current Log Configuration, then select the Log Configuration File you want to set for the log, and then click the Set Configuration button.

These are the default log configuration files you can choose between by selecting different Log Configuration Files:

Log Configuration File	Description
log4j-debug-soap.properties	The Server Log logs detailed SOAP information in addition to all the debug information from **log4j-debug.properties**. **Note:** Before contacting Spotfire Support, try to collect logs on this log level if you have a Spotfire Server issue.
log4j-debug-with-console-properties	The Server Log logs detailed debug information as well as warnings, errors and other information. The SQL Log logs detailed SQL information. If the server is started from a command prompt or shell, the output to the command prompt or shell is also included in the Server Log.
log4j-debug.properties	The Server Log logs detailed debug information as well as warnings, errors and other information. The SQL Log logs detailed SQL information. If the server is started from a command prompt or shell, the output to the command prompt or shell is also included in the Server Log.
log4j-minimal.properties	The Server Log only logs errors, and the SQL Log will be deactivated.

log4j-tools.properties The log configuration file for the configuration tool. Do not use it for Spotfire Server.

log4j-trace-soap.properties The Server Log logs extremely low-level SOAP information including the debug information from **log4j-trace.properties**.

log4j-trace.properties The Server Log logs extremely low-level debug information including the debug information from **log4j-debug.properties**.

log4j.properties The default setting. The Server Log logs warnings, errors and basic information. The SQL Log logs basic SQL information.

Note: Starting with Spotfire Server 3.3, the log configuration set on this page will persist after a Spotfire Server restart.

Note: Do not use any of the debug configurations for continuous server use as this will impact the performance of the server.

Configuring Logging:

If you want to configure logging in other ways than the default configurations, you can create your own Log Configuration File using standard Log4j syntax (refer to **http://logging.apache.org/log4j/1.2/manual.html**). This can for instance be used to change when log files are rolled (see above).

Placing a new log4j configuration file with the name matching the pattern **log4j*.properties** in the **<installation dir>/tomcat/webapps/spotfire/WEB-INF** directory, will cause it to appear in the drop-down list among the other Log Configuration Files.

Exporting Log Files

By clicking on the Export Log File button, you can save the current log file to disk.

Logging Out

For security reasons, always make sure to exit your browser when logging out of Server Logs and Diagnostics. This makes sure no session cookies are retained.

SF.2 server.xml

Spotfire Server is implemented as a Tomcat web application. For this reason, it uses a standard Tomcat web application configuration file, **server.xml**, to store information it needs when starting. This file is stored in **<installation dir>/tomcat/conf/>**.

You should only need to make changes to this file if you need to change port numbers after installation, or if you need to tweak Tomcat behavior. that each Spotfire Server in a cluster has a **server.xml** file. Therefore, if you need to make changes to it, you need to make those changes to all the servers in the cluster.

Note: The variable **[SpotfirePort]** is set when running the Spotfire Server installer. The variable **[ServerHostname]-srv** is automatically set by the installer by adding the strings **-srv** to the server's hostname. Also note that this variable must not have any characters that need escaping, such as "**.**", for instance **spotfireserver1.example.com**.

For details about the **server.xml** syntax, refer to Apache Tomcat documentation at http://tomcat.apache.org/

SF.3 bootstrap.xml

The bootstrap configuration file contains the basic information the server needs to bootstrap itself so that it can connect to the Spotfire database and retrieve its configuration.

The bootstrap configuration file is created by running the **bootstrap** command. The file must be created in the **<installation directory>\tomcat\webapps\spotfire\WEB-INF** directory (Windows) or the **<installation directory>/tomcat/webapps/spotfire/WEB-INF** directory (Unix). When specifying an alternative bootstrap configuration file path to the **bootstrap** command, the generated file must be manually copied to this directory before it can be used by the server. The file must also be named **bootstrap.xml**.

▶ To create the bootstrap.xml file:

Use the **bootstrap** command (page 195).

The bootstrap configuration file has the format displayed below:

```
<bootstrap>
  <server-name>...</server-name>
  <server>
    <driver-class>...</driver-class>
    <database-url>...</database-url>
    <username>...</username>
    <password>...</password>
  </server>
  <config-tool>
    <driver-class>...</driver-class>
    <database-url>...</database-url>
    <username>...</username>
    <password>...</password>
  </config-tool>
  <server-name>...</server-name>
  <encryption-password>...</encryption-password>
</bootstrap>
```

The **<config-tool>** section is optional and is not required for running the server itself. It is only required for using the commands to access the database. If the commands are not to be used on a specific server, they can easily be disabled by removing this section.

The database password stored in this section is protected by a special configuration tool password that is specified when creating the **bootstrap.xml** file using the **bootstrap** command. This tool password must be specified whenever running a command that accesses the database. Please note that the tool password is not related to any administrator user account within the server application itself.

The **<server-name>** section contains the server name which is used for identifying the server, for example when specifying server specific configuration.

The **<encryption-password>** section is optional. If specified it will contain a password to be used for encrypting other passwords stored in the database. If not set a static password will be used. that the same password must be configured for all servers in a cluster.

SF.4 krb5.conf

This file contains settings for Kerberos. The unmodified version of the file is presented first, then a version with example values:

Unmodified File

```
[libdefaults]
 default_realm = MYDOMAIN
 default_keytab_name = spotfire.keytab
 default_tkt_enctypes = rc4-hmac
 default_tgs_enctypes = rc4-hmac

[realms]
 MYDOMAIN = {
   kdc = mydc.mydomain
   admin_server = mydc.mydomain
   default_domain = mydomain
 }

[domain_realm]
 .mydomain = MYDOMAIN
 mydomain = MYDOMAIN

[appdefaults]
 autologin = true
 forward = true
 forwardable = true
 encrypt = true
```

File with Example Values

```
[libdefaults]
 default_realm = RESEARCH.EXAMPLE.COM
 default_keytab_name = spotfire.keytab
 default_tkt_enctypes = rc4-hmac
 default_tgs_enctypes = rc4-hmac
[realms]
 RESEARCH.EXAMPLE.COM = {
   kdc = example-dc.research.example.com
   admin_server = example-dc.research.example.com
   default_domain = research.example.com
}
[domain_realm]
 .research.example.com = RESEARCH.EXAMPLE.COM
 research.example.com = RESEARCH.EXAMPLE.COM
[appdefaults]
 autologin = true
 forward = true
 forwardable = true
 encrypt = true
```

Reference: Configuration References

This reference provides context to the Spotfire Server database connection and parameters for the connection pool. Examples of connection URLs and drivers are also provided.

CR.1 Server Bootstrapping and Database Connection Pool Configuration

The Spotfire database holds all user data and most of the configuration for the Spotfire system. In order to connect to the Spotfire database that Spotfire Server uses a database connection pool. See “Database Connectivity” on page 179.

The **bootstrap.xml** file contains the information the server needs to connect to the Spotfire database and retrieve the configuration. Refer to “bootstrap.xml” on page 176.

After the server has retrieved the configuration from the database it will re-initialize its database connection pool using information from both the **bootstrap.xml** file, which is present on each server, and any database configuration set for the entire cluster, stored as part of the database persisted server configuration.

▸ To configure the common database configuration:

Use the commands **modify-db-config** (page 267) and **set-db-config** (page 275).

Database Connectivity

Spotfire Server has a database connection pool implementation that is used for two things:

- To connect to the Spotfire database
- To connect to JDBC compliant data sources through Information Services

Each connection pool (either for Spotfire Server itself or for fetching data) has many parameters, where the following are of general interest:

- The **driver-class** parameter contains the JDBC driver class name. See “Database Connection URL Examples” on page 181.
- The **url** parameter contains the JDBC connection URL. See “Database Connection URL Examples” on page 181.
- The **username** parameter contains the name of the database user to connect as, if applicable.
- The **password** parameter contains the password for the specified database user, if applicable. The password is always encrypted and must therefore be set using the **bootstrap** command (page 195). It cannot be set manually.
- The **min-connections** parameter contains the minimum number of allocated connections.

- The **max-connections** parameter contains the maximum number of allocated connections. Depending on the pooling scheme, the total number of connections created by the server may be higher than the value of this parameter during high load, but all such extra connections will automatically be closed when the load decreases. By setting this parameter to zero or a negative value, connection pooling is effectively disabled and new connections will be continuously created, whenever needed.
- The **pooling-scheme** parameter defines the connection pooling algorithm to be used. There are two possible connection pooling algorithms that determine the way the connection pool operates, “DYNAMIC” and “WAIT”. The “WAIT” algorithm is default. The pooling algorithms are described in detail below.

The “DYNAMIC” pooling scheme

When initialized, the connection pool creates a number of idle database connections equal to the **min-connections** parameter. When the connection pool receives a request for a database connection, it checks if the pool contains any idle connections and uses one of those, if available. If there are no idle connections in the pool, it automatically creates a new database connection. There is no upper limit for how many connections a connection pool can have open at the same time.

Idle connections in the pool eventually time out if they aren’t used. The **connection-timeout** parameter defines how long time (given in seconds) a connection can stay idle in the connection pool before being closed and discarded.

The “WAIT” pooling scheme

When initialized, the connection pool creates a number of idle database connections equal to the **min-connections** parameter. When the connection pool receives a request for a database connection, it checks if the pool contains any idle connections and uses one of those, if available:

- If there are no idle connections in the pool and the number of already open connections is less than the **max-connections** parameter, it creates a new database connection.
- If the number of already open connections is equal to the **max-connections** parameter, it waits for an active connection to be returned to the pool. If the request cannot be fulfilled within a number of seconds equal to the **login-timeout** parameter, the request times out. In the server logs entries like this appear: “`Timeout while waiting for database connection after 10 seconds`”.

Thus, in WAIT mode, the connection pool can never have more open (active or idle) connections than the value of the **max-connections** parameter. Whenever a database connection is returned, it is put in the pool of idle connections, unless it is used immediately to fulfill an already waiting request.

Idle connections in the pool eventually time out if they aren’t used. The **connection-timeout** parameter defines how long time (given in seconds) a connection can stay idle in the connection pool before being closed and discarded.

CR.2 Database Connection URL Examples

This section contains database connection URL examples for connecting to the Spotfire database.

Databases and JDBC drivers supported for running Spotfire Server

- Oracle (DataDirect Driver)

  Driver name: **tibcosoftwareinc.jdbc.oracle.OracleDriver**

- Oracle (Oracle JDBC Thin Driver, **ojdbc6.jar**)

  Driver name: **oracle.jdbc.OracleDriver**

- Microsoft SQL Server (DataDirect Driver)

  Driver name: **tibcosoftwareinc.jdbc.sqlserver.SQLServerDriver**

- Microsoft SQL Server (Microsoft JDBC Driver, **sqljdbc4.jar**)

  Driver name: **com.microsoft.sqlserver.jdbc.SQLServerDriver**

- Microsoft SQL Server (jTDS JDBC Driver, **jtds.jar**)

  Driver name: **net.sourceforge.jtds.jdbc.Driver**

Database Connection URL Components

Component	Description
API	Specifies which API to use. This is always jdbc.
Database Driver	Specifies which database driver to use to connect to the database. Default **tibcospotfireinc**, which will use the Spotfire DataDirect driver. If you have installed a different driver, you may provide this here.
Server Type	Specifies the type of database server. Either sqlserver or oracle. **Note:** Server Type is only applicable when using the DataDirect driver.
Hostname	Specifies the hostname of the database server.
Port	Specifies the port which the database server listens to; for example 1433.
Database Name or SID	Specifies the name (MSSQL) or SID (Oracle) that defines your Spotfire database.
Options	Specifies further options, separated with semicolons. Only necessary if you need to set something specific for your database server. This may include information such as a named Instance in an MSSQL server, for example. See example below.

Database URL Examples

For the different supported database drivers the URL components become:

Oracle (DataDirect Driver):

[API]:[DBDriver]:[ServerType]://[Hostname]:[Port];SID=[SID]

Example:

jdbc:tibcospotfireinc:oracle://dbsrv.example.com:1433;SID=spotfire_server

Oracle (Vendor Driver, **ojdbc6.jar**):

[API]:[DBDriver]:[DriverType]://[Hostname]:[Port]:SID

Example:

jdbc:oracle:thin:@dbsrv.example.com:1521:orcl

Microsoft SQL Server (DataDirect Driver)

[API]:[DBDriver]:[ServerType]://[Hostname]:[Port];DatabaseName=[DBName]

Example:

jdbc:tibcosoftwareinc:sqlserver://dbsrv.example.com:1433;DatabaseName= spotfire_server

Example using Integrated Authentication:

jdbc:tibcosoftwareinc:sqlserver://dbsrv.example.com:1433;DatabaseName= spotfire_server;AuthenticationMethod=ntlm;LoadLibraryPath=c:/tibco/tss/6.5.0/tomcat/ lib

Note: Make sure that the LoadLibraryPath has the correct path to the **tomcat/lib** directory in Spotfire Server installation directory.

Microsoft SQL Server (Vendor Driver, **sqljdbc4.jar**)

[API]:[DBDriver]://[Hostname]:[Port];DatabaseName=[DBName]

Example:

jdbc:sqlserver://dbsrv.example.com:1433;DatabaseName=spotfire_server;selectMethod= cursor

Example: Making sure that the driver always returns will prevent infinite waits during adverse conditions

jdbc:sqlserver://dbsrv.example.com:1433;DatabaseName=spotfire_server;lockTimeout= <X, where X is a good value>

Note: Due to a restriction in the vendor Microsoft SQL Server driver, you may need to add the option `responseBuffering=adaptive` to your connection string. This is necessary if you are going to store large analysis files in the library.

Example: Using **responseBuffering=adaptive**:

jdbc:sqlserver://dbsrv.example.com:1433;databaseName=spotfire_server;selectMethod=cursor;responseBuffering=adaptive

Example: Using Integrated Authentication:

jdbc:sqlserver://dbsrv.example.com:1433;DatabaseName=spotfire_server;selectMethod=cursor;integratedSecurity=true;

Note: For integrated authentication to work, you must place the file sqljdbc_auth.dll in a folder in the system path, such as C:\Windows\System32. This file is included with the vendor drivers from Microsoft.

jTDS (Vendor Driver, **jtds.jar**)

jdbc:jtds:sqlserver://[Hostname]:[Port]/[DatabaseName]

Example:

jdbc:jtds:sqlserver://dbsrv.example.com:1433/spotfire_server

Microsoft SQL Server with Named Instance
If your database server uses a named Instance, you must add an option to the end of the connection string to identify it. Note that this option is not limited to the jTDS driver. You can add this option regardless of what driver you are using.

jdbc:jtds:sqlserver://[Hostname]:[Port]/[DatabaseName];instance=[InstanceName]

Example:

jdbc:jtds:sqlserver://dbsrv.example.com:1433/spotfire_server;instance=FirstInstance

Note: To use other JDBC drivers than the DataDirect ones, you need to install them onto each Spotfire Server. Refer to the section “Install Database Drivers” on page 27 for more information about this.

Reference: Load Balancing Reference Implementation

Spotfire Server can be deployed to multiple machines, creating a server cluster. This can be used to achieve failover and to balance load between Spotfire Servers. Each Spotfire Server in a cluster communicates with the Same Spotfire database.

While it is possible to allow users to connect directly to any of the Spotfire Servers, a more common approach is to use a load balancer. A load balancer is a computer that acts as a front for the Spotfire Servers. The clients connect to the load balancer, which then redirects their communication to an available Spotfire Server. The client has to communicate with the same server during the entire session; that is, sticky sessions or session affinity is maintained. The load balancer must support session affinity and be able to detect if a Spotfire Server becomes available or unavailable.

It is possible to use any load balancing technology that supports session affinity. Spotfire Server is implemented using Apache Tomcat, which means that it supports load balancing via AJP (Apache JServ Protocol).

This chapter is a reference implementation for a load balancing setup using AJP and a load balancer implementation using Apache HTTP Server with the **mod_jk** module. The Apache HTTP Server is not supported by TIBCO.

If you intend to use a login method that authenticates users with an external directory, this may have implications on how the load balancer should be set up.

The procedure is to set up one Spotfire Server to be load balanced. When traffic is being forwarded as it should, others can be added.

LB.1 Prerequisites

Spotfire Server

One or more Spotfire Servers installed. It may be a good idea to start with one server and then add more later.

Load Balancer

A load balancer that supports session affinity. If this is the Apache HTTP Server, it will need:

- The **mod_jk** module installed and enabled.
- Optional: If NTLM authentication is used, the **mod_auth_sspi** module installed and enabled.

LB.2 Spotfire Server Configuration

To enable Spotfire Server to communicate with a load balancer using the AJP protocol the **<server installation directory>/tomcat/conf/server.xml** file has been edited.

The following connector section has been uncommented:

```
<!-- Enable this connector if you want to use a load balancer that
supports the Apache JServ Protocol -->
  <Connector port="8009"
     protocol="AJP/1.3"
     packetSize="65536"
     URIEncoding="UTF-8"/>
```

Optionally, to prevents clients from connecting to Spotfire Server directly, forcing them to use the load balancer, HTTP communication can be turned off by commenting out the following connector section:

```
  <Connector port="80"
     maxHttpHeaderSize="16384"
     connectionTimeout="30000"
     enableLookups="false"
     URIEncoding="UTF-8"
     disableUploadTimeout="true"
     server="TIBCO Spotfire Server"/>
```

LB.3 Load Balancer Configuration

The load balancer has been configured to be able to find and communicate with Spotfire Servers.

1 Apache httpd has been installed.

2 The **mod_jk** module has been installed. Refer to the Apache httpd manual for details.

3 The following to **workers.properties** have been added. You may need to create this file:

```
# Define worker list
# (All workers with additional exposed applications must also be added
here,
#  and don't forget to add the corresponding JkMount option in
mod_jk.conf!)
worker.list=jkstatus, loadbalancer
# Example: the /admin application on worker1 should be exposed through
the load balancer
#worker.list=jkstatus, loadbalancer, worker1

# Set status
worker.jkstatus.type=status

# Set properties for the load balancer
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=worker1, worker2
worker.loadbalancer.sticky_session=true
worker.loadbalancer.method=Session
```

```
# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=[SpotfireServer1Hostname]
worker.worker1.port=8009
worker.worker1.max_packet_size=65536
worker.worker1.lbfactor=1
worker.worker1.route=[SpotfireServer1Hostname]-srv

# Set properties for worker2 (ajp13)
worker.worker2.type=ajp13
worker.worker2.host=[SpotfireServer2Hostname]
worker.worker2.port=8009
worker.worker2.max_packet_size=65536
worker.worker2.lbfactor=1
worker.worker2.route=[SpotfireServer2Hostname]-srv
```

Change **[SpotfireServer1Hostname]** to the hostname or IP address of your first Spotfire Server, **[SpotfireServer2Hostname]** to the name of your second Spotfire Server, etc. The AJP route is automatically set to **[SpotfireServerHostname]-srv** on Spotfire Server end at installation, that is the hostname of the server suffixed by **-srv**. The following has been added to the **mod_jk.conf** file. You may need to create this file:

```
# Load the mod_jk module
LoadModule jk_module modules/mod_jk.so

# Load the workers configuration
JkWorkersFile conf/workers.properties

# The mod_jk module's log file
JkLogFile logs/mod_jk.log

# The mod_jk module's log level (trace, debug, info, warn, error)
JkLogLevel info

# Let the load balancer worker handle all requests to the TSS web
applications
JkMount /spotfire loadbalancer
JkMount /spotfire/* loadbalancer

# Define Apache environment variables to be exported by mod_jk to
Tomcat web applications
JkEnvVar REMOTE_USER
JkEnvVar SSL_CLIENT_CERT
#JkEnvVar SSL_CLIENT_CERT_CHAIN
#JkEnvVar SSL_CLIENT_S_DN
#JkEnvVar SSL_CLIENT_S_DN_CN
```

4 It is verified that the Apache httpd configuration includes the file **mod_jk.conf**.

5 The Apache httpd is restarted and checked for startup errors.

6 It is verified that it is possible to connect to each server using both HTTP on the ports defined in the installation process and using AJP on port 8009.

7 A higher level of security can be achieved by making the load balancer authenticate when it talks to Spotfire Servers. See “Setting up HTTPS” on page 189 for details.

LB.4 Load Balancer AJP Keyword Restriction

Apart from Load Balancer Authentication, it is also possible to set up a AJP Connector secret keyword for the load balancers to use to authenticate with Spotfire Servers. This is a secret keyword that the load balancers and Spotfire Servers all know. You set this up by doing the following:

1 Add the keyword to all Spotfire Servers.
In the Spotfire Server **server.xml**, within the section **<Service name="Spotfire">**, find the Connector configuration: **<Connector port="8009" protocol="AJP/1.3" packetSize="65536"/>**

Add the keyword definition:

```
<Connector port="8009"
protocol="AJP/1.3"
packetSize="65536"
request.useSecret="true"
request.secret="SecretKeyword" />
```

2 Then, add the keyword to the **worker,properties** file on the load balancer. Above the properties for the individual workers, add a keyword for all the workers to use:

```
# Enable secret keyword
worker.loadbalancer.secret="SecretKeyword"
```

When this is set up, the Spotfire Server will only accept AJP connections from load balancers that know the secret keyword.

LB.5 Kerberos Authentication

In a clustered environment where Kerberos authentication is used to authenticate users, the load balancer forwards all Kerberos authentication information to the Spotfire Servers. No configuration on the load balancer is needed, but there are certain considerations that must be taken into account when Kerberos authentication is set up:

- Two Service Principal Names must be created for each Spotfire Server as well as for the load balancer.
- One keytab file must be created. This must use the fully qualified Service Principal Name of the load balancer.
- This keytab file must be copied to each Spotfire Server.
- When Kerberos authentication is set up, the fully qualified Service Principal Name of the load balancer must be provided.

LB.6 X.509 Client Certificate Authentication

When using X.509 Client Certificate authentication in a clustered environment, the clients see the load balancer as the server, and the load balancer must therefore be provided and configured with a server certificate and its private key. The load balancer also needs to be provided and configured with the CA certificate that was used to issue the server certificate. See the section “Setting up HTTPS” on page 189 and “Configuring X.509 Client Certificates” on page 191.

On Spotfire Server the Delegate authentication method must be enabled for the load balancer, so that the load balancer can forward the username extracted from the client certificates. See “External Authentication Method” on page 73.

LB.7 Setting up HTTPS

In a clustered environment, the clients see the load balancer as the server. Therefore, in order to secure the communication in the Spotfire system, using HTTPS, the load balancer needs to be configured to do this. This is achieved by obtaining (or creating) a server certificate for the load balancer and installing it on the load balancer. You may need to convert it first.

Note: You cannot use a server certificate created for a Spotfire server. A server certificate must always be created for the computer it is intended for.

The following instructions assume that you are acquainted with the Apache httpd and its configuration files. It should be seen as an overview of how HTTPS is setup for use in load balancing a Spotfire system, not as a tutorial on Apache httpd. For more information, please refer to the Apache httpd manual.

1. Install Apache httpd with ssl support and the mod_ssl.so and mod_jk modules.
2. Create a self-signed server certificate.
3. If needed, convert the certificate to a format readable by the load balancer.
4. Store the converted certificate in the Apache conf directory.
5. Configure Apache to use the certificate files.
6. Make your clients trust the CA certificate.

Install Apache httpd with SSL Support and the mod_ssl.so and mod_jk modules

For exact instructions on how to install Apache httpd, see the Apache manual.

If you are using an Apache installer, you might be presented with the option of creating a self-signed server certificate from within the installer, and have Apache automatically configured to use this server certificate. If this is the case, you can skip to the step “Make your clients trust the CA certificate”. If you are not using an automatic installer, continue.

Obtain a Certificate

You must obtain a certificate to use with the Apache httpd. This can be obtained from a commercial Certificate Authority, or by creating one yourself. Please turn to your provider for information about how to do this.

After obtaining this certificate, save it to file and transfer it to the load balancer.

Convert the Certificate to a Format Readable by the Load Balancer

The certificate must be in the Base 64-encoded DER format (PEM) format for Apache httpd to be able to read it. If the certificate is created with Microsoft Certificate Services, it is in the PKCS #12 format. To convert it, use the openssl command on the load balancer. (If this is not installed, refer to http://openssl.org/ or your OS manual for instructions on how to install it.)

```
openssl pkcs12 -in server.pfx -out server.pem
```

Next, the public key in the certificate must be extracted from the converted certificate:

```
openssl x509 -in server.pem -out server_cert.pem
```

Finally, the private key in the certificate must be extracted from the converted certificate:

```
openssl rsa -in server.pem -out server_key.pem
```

These commands will provide you with three files: **server.pem**, **server_cert.pem**, and **server_key.pem**. You will only need the two latter files.

You also need the CA certificate on the load balancer in the PEM format. If you are using a self-signed certificate, the CA certificate should be available for download from the same source, usually under “Trusted Root Certification Authorities” or similar. If needed, convert the CA certificate to PEM format using the convert command above. You do not need to extract anything from it.

Store the Converted Certificate Files in the Apache Conf Directory

Copy all the files created above to the directory **<apache httpd dir>/conf**.

Configure Apache httpd to Use the Certificate Files

Add the following lines to the Apache httpd configuration (for instance, to the load balancer’s virtual host):

```
# Configure SSL
SSLEngine On
SSLCertificateFile "conf/server_cert.pem"
SSLCertificateKeyFile "conf/server_key.pem"
SSLCACertificateFile "conf/cacert.pem"
SSLOptions +StdEnvVars +ExportCertData
```

Your Apache httpd should now communicate using the HTTPS protocol.

Make Your Clients Trust the CA Certificate

If you have obtained a CA Certificate from a commercial CA, your clients probably already trust it. If you have created it yourself, please refer to your CA software documentation on how to get clients to trust it.

LB.8 Configuring X.509 Client Certificates

In a load balanced environment, where X.509 Client Certificate authentication is to be used, the load balancer needs to be aware of the situation and forward the client certificates to the Spotfire Servers.

The following instructions assume that you are acquainted with the Apache httpd and its configuration files. It should be seen as an overview of how HTTPS is setup for use in load balancing a Spotfire system, not as a tutorial on Apache httpd. For more information, please refer to the Apache httpd manual

1. Configure the Spotfire system to use X.509 Client Certificate authentication. See "Authentication Using X.509 Client Certificates" on page 72.
2. Configure Apache httpd to communicate using the HTTPS protocol. See the section "Setting up HTTPS" on page 189.
3. Configure Apache httpd to require and forward X.509 client certificates.
4. Configure mod_jk to forward X.509 client certificates.

Configure Apache httpd to Require and Forward X.509 Client Certificates

Add the following lines to the Apache httpd configuration (for instance, to the load balancer's virtual host, where the HTTPS configuration was added):

```
# Configure client cert
SSLVerifyClient require
SSLVerifyDepth 1
SSLUserName SSL_CLIENT_S_DN_CN
# Configure mod_jk directives
JkMountCopy On
JkOptions +ForwardKeySize +ForwardSSLCertChain
```

Configure mod_jk to Forward X.509 Client Certificates

Add the following to the mod_jk configuration (typically, a file called mod_jk.conf included by httpd.conf or httpd-ssl.conf)

```
JkEnvVar SSL_CLIENT_CERT
```

You should now have a load balancer that requires and forwards X.509 Client Certificates.

LB.9 Shared Disk Location

From the Library Administration tool found in the Spotfire client, it is possible to import and export content to and from the library. The import and export files are stored in a folder specified in the Spotfire Server configuration.

In a clustered environment, there is no way of being certain which server the client is communicating with. Therefore, steps must be taken to ensure that the import and export files are always stored in the same folder.

One method is to use Windows shared folder technology, and set the location of the import and export folder to a folder that is shared with all Spotfire Servers. There is also a method in which you can configure the load balancer to always redirect import and export requests to the same Spotfire Server.

To set this up using Apache httpd as a load balancer, you need to add the following to the mod_jk configuration (such as in the file mod_jk.conf):

```
JkUnmount /spotfire/ws/LibraryImportExportService loadbalancer
JkUnmount /spotfire/ws/LibraryImportExportService/* loadbalancer
JkMount /spotfire/ws/LibraryImportExportService worker1
JkMount /spotfire/ws/LibraryImportExportService/* worker1
```

worker1 is the Spotfire Server where import and export files are to be stored. Then, you need to add the worker1 to the list of workers in the workers.properties file:

worker.list=jkstatus, loadbalancer, worker1

With this setup, all import and export calls from the Library Administration tool will terminate on the Spotfire Server worker1.

Reference: Commands

Commands are listed alphabetically. Refer to "Available Configuration Commands" on page 45 for an easily reviewed functional command grouping and "Authentication and User Directory" on page 55 for the procedure to configure using commands.

Synopsis:Optional.

- Angle brackets (<>) indicate mandatory arguments,
- Square brackets (`[]`) indicate optional arguments.
- Arguments can normally be specified in two different formats. For example, the **max cache size** argument as: **--max-cache-size=<value>** or **-m <value>**.

A negative value must be preceded by a backslash in the later mentioned argument format. For example **-m \-7**.

add-ds-template

Adds a new data source template.

SYNOPSIS

```
add-ds-template [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-n value | --name=value> [-e <true|false>
  | --enabled=<true|false>] <template definition file>
```

OVERVIEW

Use this command to add a new data source template used by Information Services. The name of the template must be unique.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-n value
--name=value
```

Required. The name of the data source template to add.

```
-e <true|false>
--enabled=<true|false>
```

Optional. Indicates whether the newly created data source template should be enabled. The default value is **false**.

```
<template definition file>
```

Required. The path to the file containing the data source template definition.

add-member

Adds a user or group as a member of a specified group.

SYNOPSIS

```
add-member [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-g value | --groupname=value> [-u value |
  --member-username=value] [-m value | --member-groupname=value]
```

OVERVIEW

Use this command to add an existing user or group as a member of another existing group.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “Bootstrap tab” on page 37.

```
-g value
--groupname=value
```

Required. The name of the group to which the member should be added. Unless the group is part of the internal **SPOTFIRE** domain, the name of the group must include the group's domain name, for example **RESEARCH\group** or **group@research.example.com**.

```
-u value
--member-username=value
```

Required, unless the **--member-groupname** argument is specified. The name of the user to add as a member of the specified group. Unless the user is part of the configured default domain, the name of the user must include the user's domain name: For example, **RESEARCH\user** or **user@research.example.com**. The **--member-username** and **--member-groupname** arguments are mutually exclusive.

```
-m value
--member-groupname=value
```

Required, unless the **--member-username** argument is specified. The name of the group to add as a member of the specified group. Unless the group is part of the internal **SPOTFIRE** domain, the name of the group must include the group's domain name: For example, **RESEARCH\group** or **group@research.example.com**. The **--member-username** and **--member-groupname** arguments are mutually exclusive.

bootstrap

Bootstraps the server by creating a new **bootstrap.xml** file containing the information needed to connect to the database.

SYNOPSIS

```
bootstrap [-f | --force] [-T | --test] [-n | --no-prompt] [-c value |
  --driver-class=value] [-d value | --database-url=value] [-u value |
  --username=value] [-p value | --password=value] [-k value |
  --kerberos-login-context=value] {-Ckey=value} [-E <true|false> |
  --enable-config-tool=<true|false>] [-t value | --tool-password=value]
  [-e value | --encryption-password=value] [-s value | --server-name=
  value] [bootstrap configuration file]
```

OVERVIEW

Use this command to create a new bootstrap configuration file.

OPTIONS

```
-f
--force
```

Optional. Indicates that the tool should overwrite any already-existing bootstrap configuration file.

```
-T
--test
```

Optional. Specifies that the tool should test the created configuration by attempting to connect to the database using the specified connection information.

```
-n
--no-prompt
```

Optional. Specifies that the tool should not prompt for missing password arguments.

```
-c value
--driver-class=value
```

Optional. The name of the JDBC driver class. The default value is **tibcosoftware-inc.jdbc.oracle.OracleDriver**.

```
-d value
--database-url=value
```

Optional. The JDBC URL to the database. Because this argument usually contains special characters, make sure to escape those characters or enclose the values between quotes. The default value is:
jdbc:tibcosoftwareinc:oracle://localhost:1521;SID=orcl

```
-u value
--username=value
```

Optional. The database account username.

```
-p value
--password=value
```

Optional. The database account password.

```
-k value
--kerberos-login-context=value
```

Optional. If you use the Kerberos protocol to log in to the database, use this argument to specify the name of the JAAS application configuration to be used for acquiring the Kerberos TGT. This JAAS application configuration must be registered with Java using a **login.config.url** parameter in the **<TSS installation directory>\jdk\jre\lib\security\java.security** (Windows) or **<TSS installation directory>/jdk/jre/lib/security/java.security** (Unix) file.

The Spotfire Server **import-jaas-config** command cannot be used for this purpose because the JAAS application configurations that are imported using this command are stored in the database, which prevents the Spotfire Server from using them for creating the initial connection to the database.

```
-Ckey=value
```

Optional. A JDBC connection property. Can be specified multiple times with different keys.

```
-E <true|false>
--enable-config-tool=<true|false>
```

Optional. If **true** (the default), the <***config-tool***> section should be created. Without this section, the configuration tool cannot be used on this computer. See the **bootstrap.xml** help topic for more information.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. Can be specified if and only if a password is given and the argument **--enable-config-tool** is set to **true** (the default).

```
-e value
--encryption-password=value
```

Optional. The password for encrypting passwords stored in the database. If you do not set this option, a static password is used. that the same password must be configured for all servers in a cluster.

```
-s value
--server-name=value
```

Optional. The server name. Used for identifying the server. For example, you can use it to specify a server-specific configuration. The default value is the fully qualified host name as determined when this command is run.

```
[bootstrap configuration file]
```

Optional. The path to the bootstrap configuration file to create. See “bootstrap.xml” on page 176.

EXAMPLES

Bootstrap the server to use an Oracle database with the bundled DataDirect JDBC driver:

```
bootstrap  --driver-class=tibcosoftwareinc.jdbc.oracle.OracleDriver  --database-url=
"jdbc:tibcosoftwareinc:oracle://server:1521;SID=spotfire"  --username=spotuser
--password=spotuser
```

Bootstrap the server to use an Oracle database with the Oracle thin JDBC driver:

```
bootstrap  --driver-class=oracle.jdbc.OracleDriver  --database-url="jdbc:oracle:thin:@
server:1521:spotfire"  --username=spotuser  --password=spotuser
```

Bootstrap the server to use a Microsoft SQL Server database with the bundled DataDirect JDBC driver:

```
bootstrap --driver-class=tibcosoftwareinc.jdbc.sqlserver.SQLServerDriver --database-url=
"jdbc:tibcosoftwareinc:sqlserver://server:1433;DatabaseName=spotfire_server"
--username=spotuser --password=spotuser
```

Bootstrap the server to use a Microsoft SQL Server database with the Microsoft JDBC driver:

```
bootstrap --driver-class=com.microsoft.sqlserver.jdbc.SQLServerDriver --database-url=
"jdbc:sqlserver://server:1433;DatabaseName=spotfire_server" --username=spotuser
--password=spotuser
```

Bootstrap the server to use a Microsoft SQL Server database with the jTDS JDBC driver:

```
bootstrap  --driver-class=net.sourceforge.jtds.jdbc.Driver  --database-url=
"jdbc:jtds:sqlserver://server:1433/spotfire_server"  --username=spotuser  --password=
spotuser
```

check-external-library

Checks for inconsistencies between external storage and the Spotfire database.

SYNOPSIS

```
check-external-library [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-t value | --tool-password=value]
```

OVERVIEW

Use this command to check the consistency between what is stored in external storage (for example, Amazon S3 or a file system), and what is stored in the Spotfire database.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

clear-join-db

Clears the default join database configuration.

SYNOPSIS

```
clear-join-db [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to clear the default join database configuration, which means that the Spotfire database is used as the default join database (the default behavior).

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

config-action-log-database-logger

Configures the user action database logger.

SYNOPSIS

```
config-action-log-database-logger [-c value | --configuration=value]
  [-b value | --bootstrap-config=value] [--driver-class=value] [-d
  value | --database-url=value] [-u value | --username=value] [-p value
  | --password=value] [--commit-period=value]
  [--wait-on-full-queue-time=value] [--wait-on-empty-queue-time=value]
  [--grace-period=value] [--pruning-period=value] [--queue-size=value]
  [--batch-size=value] [--thread-pool-size=value] [--workers=value]
  [--block-on-full-queue=<true|false>][--prioritized-categories=value]
  [--monitoring-retention-span=value] [--monitoring-average-period=
  value]
```

OVERVIEW

This command configures the user action database logger.

OPTIONS

`-c value`

`--configuration=value`

Optional. The path to the server configuration file. The default value is **configuration.xml**.

`-b value`

`--bootstrap-config=value`

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

`--driver-class=value`

Optional. The name of the JDBC driver class.

`-d value`

`--database-url=value`

Optional. The JDBC URL to the database. Because this argument usually contains special characters, be sure to escape those characters or enclose the values between quotes.

`-u value`

`--username=value`

Optional. The database account username.

`-p value`

`--password=value`

Optional. The database account password.

`--commit-period=value`

Optional. The frequency (in seconds) that log events should be committed from the queue to the database when the queue is not full.

`--wait-on-full-queue-time=value`

Optional. The time (in milliseconds) to wait before retrying to place a new log event on the queue after being rejected by a full queue.

`--wait-on-empty-queue-time=value`

Optional. Sets the time (in milliseconds) to wait before trying to create a batch from the queue after an empty queue has been encountered.

`--grace-period=value`

Optional. The grace period for the database logger (in seconds). This is the period that the database logger is given at server shutdown to move all items from the queue to the database.

`--pruning-period=value`

Optional. The maximum time (in hours) that logged items are kept in the database. Pruning takes place at server startup, and then at one hour intervals, when all items older than the here-specified number of hours are deleted. To disable pruning, set this argument to **0**.

`--queue-size=value`

Optional. The maximum number of log events in the queue.

`--batch-size=value`

Optional. The number of log events that should be moved from the queue to the database in each batch insert.

`--thread-pool-size=value`

Optional. The number of threads available for the batch insert workers.

`--workers=value`

Optional. The maximum number of batch insert workers at any given time.

`--block-on-full-queue=<true|false>`

Optional. Specifies whether placing a log event on the queue should be allowed to be blocked indefinitely if the queue is full.

`--prioritized-categories=value`

Optional. A comma separated list of log categories that should have higher priority in the queue.

`--monitoring-retention-span=value`

Optional. The length of time monitoring entries should be saved before they get crunched into averages.

`--monitoring-average-period=value`

Optional. The period between two averaged measurements.

config-action-log-web-service

Configures the action log web service.

SYNOPSIS

```
config-action-log-web-service [-c value | --configuration=value] [-b
  value | --bootstrap-config=value] [--categories=value]
  [--allowedHosts=value]
```

OVERVIEW

This command configures the user action log web service.

OPTIONS

`-c value`
`--configuration=value`

Optional. The path to the server configuration file. The default value is **configuration.xml**.

`-b value`
`--bootstrap-config=value`

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

`--categories=value`

Optional. A comma-separated list of categories that should be allowed to log through the web service. To enable all categories, specify **all**.

`--allowedHosts=value`

Optional. A regular expression that sets the hosts allowed to use the logger web service. To enable all hosts, specify **.***.

config-action-logger

Configures the user action logger.

SYNOPSIS

```
config-action-logger [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [--categories=value]
  [--file-logging-enabled=<true|false>] [--database-logging-enabled=
  <true|false>][--monitoring-period=value]
```

OVERVIEW

This command configures the user action logger.

OPTIONS

`-c value`
`--configuration=value`

Optional. The path to the server configuration file. The default value is **configuration.xml**.

`-b value`
`--bootstrap-config=value`

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

`--categories=value`

Optional. A comma-separated list of the categories that should be logged by the user action logger. To enable logging for all categories, specify **all**.

`--file-logging-enabled=<true|false>`

Optional. Specifies whether the user action logger should log to file.

`--database-logging-enabled=<true|false>`

Optional. Specifies whether the user action logger should log to database.

`--monitoring-period=value`

Optional. Specifies how often monitoring measures are reported

config-attachment-manager

Configures the Attachment Manager.

SYNOPSIS

```
config-attachment-manager [-c value | --configuration=value] [-b value
  | --bootstrap-config=value] [-e value | --max-cache-expiration-time=
  value] [-m value | --max-cache-size=value] [-E <true|false> |
  --encryption-enabled=<true|false>] [-k value | --encryption-key-size=
  value]
```

OVERVIEW

Use this command to configure the Attachment Manager, which handles data transfer (for instance Library downloads and uploads) to and from the Spotfire Server.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-e value
--max-cache-expiration-time=value
```

Optional. The maximum idle time (in seconds), after which cache entries are evicted. Setting this parameter to a negative value disables the cache. The default value is **86400**.

```
-m value
--max-cache-size=value
```

Optional. The maximum amount of disk space (in megabytes) used by the cache. Setting this parameter to a negative value disables the cache. The default value is **10240**.

```
-E <true|false>
--encryption-enabled=<true|false>
```

Optional. Specifies whether the encryption of temp files is enabled. The default value is **true**.

```
-k value
--encryption-key-size=value
```

Optional. The size of the encryption key used when encrypting temp files. The default value is **128**.

config-auth

Configures authentication mode and default domain.

SYNOPSIS

```
config-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-a value | --auth-method=value] [-d |
  --jaas-database] [-l | --jaas-ldap] [-w | --jaas-windows] [-j value |
  --jaas-custom=value] [-D value | --default-domain=value] [-p <true|
  false> | --parse-user-and-domain-name=<true|false>]
```

OVERVIEW

Use this command to configure the authentication mode and to set the default domain.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-a value
--auth-method=value
```

Optional. The authentication method to use. The following methods are supported: **BASIC**, **CLIENT_CERT**, **NTLM**, **Kerberos**, and **External**. The names can be specified in either upper or lower case.

```
-d
--jaas-database
```

Optional. Use the Spotfire database authentication source, as configured in the Spotfire-DBLogin JAAS application configuration. This option is permitted only when using the BASIC authentication method. Also, it is mutually exclusive with all other options related to BASIC authentication sources.

```
-l
--jaas-ldap
```

Optional. Use the LDAP authentication source, as configured in the SpotfireLDAP JAAS application configuration. This option is permitted only when using the BASIC authentication method. Also, it is mutually exclusive with all other options related to BASIC authentication sources.

```
-w
--jaas-windows
```

Optional. Use the Windows NT authentication source, as configured in the SpotfireWindows JAAS application configuration. This option is permitted only when using the BASIC authentication method. Also, it is mutually exclusive with all other options related to BASIC authentication sources.

```
-j value
--jaas-custom=value
```

Optional. Use the custom JAAS application configuration with the specified name. This option is permitted only when using the BASIC authentication method. Also, it is mutually exclusive with all other options related to BASIC authentication sources.

```
-D value
--default-domain=value
```

Optional. The name of the default domain. A user belonging to the default domain need not specify domain name as part of his or her username when logging in to the server. The default value is **SPOTFIRE**, which is the name of the domain used when running the User Directory in database mode.

```
-p <true|false>
--parse-user-and-domain-name=<true|false>
```

Optional. Indicates whether the username consists of both a user and a domain part that should be parsed. We recommend avoiding changing the default value of **true**, except when you are running the User Directory in database mode, and the usernames are in either NetBIOS name format (domain\user) or email name format (user@ domain).

config-auth-filter

Configures the Authentication Filter.

SYNOPSIS

```
config-auth-filter [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-f value | --filter-class=value] {-Ikey=
  value}
```

OVERVIEW

Use this command to configure a custom Authentication Filter.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-f value
--filter-class=value
```

Optional. The fully-qualified name of a class implementing the **javax.servlet.Filter** interface.

```
-Ikey=value
```

Optional. The initialization parameters provided to the filter when the **init(FilterConfig)** method is called. Can be specified multiple times with different keys.

Example

To set the initialization parameter 'debug' to 'true':

-Idebug=true

config-basic-database-auth

Configures the Spotfire Database authentication source to use the BASIC authentication method.

SYNOPSIS

```
config-basic-database-auth [-c value | --configuration=value] [-b value
  | --bootstrap-config=value] [-p <true|false> |
  --parse-user-and-domain-name=<true|false>]
```

OVERVIEW

Use this command to configure the Spotfire Database authentication source to use the BASIC authentication method. The configuration is stored in the SpotfireDatabase JAAS application configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-p <true|false>
--parse-user-and-domain-name=<true|false>
```

Optional. This argument is deprecated and is ignored. Use the **config-auth** command to set the global configuration property.

config-basic-ldap-auth

Configures the LDAP authentication source for use with the BASIC authentication method.

SYNOPSIS

```
config-basic-ldap-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-l value | --ldap-configs=value] [-w
  <true|false> | --enable-wildcard-domain=<true|false>]
```

OVERVIEW

Use this command to configure the LDAP authentication source to use the BASIC authentication method. The configuration is stored in the SpotfireLDAP JAAS application configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-l value
--ldap-configs=value
```

Optional. A comma-separated list of LDAP configuration references. All referenced LDAP configurations must already exist. To create a new LDAP configuration, use the **create-ldap-config** command. When specifying more than one reference, make sure to enclose the list of references in double quotes.

```
-w <true|false>
--enable-wildcard-domain=<true|false>
```

Optional. Indicates whether the server should attempt to authenticate the user in all domains until an authentication attempt succeeds whenever the user omits the domain name in the account name credential.

config-basic-windows-auth

Configures the Windows NT authentication source to use the BASIC authentication method.

SYNOPSIS

```
config-basic-windows-auth [-c value | --configuration=value] [-b value
  | --bootstrap-config=value] [-d value | --domains=value] [-w <true|
  false> | --enable-wildcard-domain=<true|false>]
```

OVERVIEW

Use this command to configure the Windows NT authentication source to use the BASIC authentication method. The configuration is stored in the SpotfireWindows JAAS application configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-d value
--domains=value
```

Optional. A comma-separated list of domain names. When specifying more than one domain name, make sure to enclose the list of names in quotes.

```
-w <true|false>
--enable-wildcard-domain=<true|false>
```

Optional. Indicates whether the server should attempt to authenticate the user in all domains until an authentication attempt succeeds whenever the user omits the domain name in the account name credential.

config-client-cert-auth

Configures the **CLIENT_CERT** authentication method.

SYNOPSIS

```
config-client-cert-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-n value | --name-attribute=value> [-d
  <true|false> | --name-attribute-contains-domain=<true|false>]
```

OVERVIEW

Use this command to configure the X.509 certificate name attribute used for the **CLIENT_CERT** authentication method.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-n value
--name-attribute=value
```

Required. The name of the attribute used to extract usernames from X.509 certificates.

Supported attributes are:

- Any attribute that can occur in the certificate subject's distinguished name (for instance "CN")
- "DN" (use the whole distinguished name)
- Any subject alternative name of type "rfc822Name", "dNSName", "directoryName", "uniformResourceIdentifier", "iPAddress" or "registeredID".

  To use a subject alternative name, make sure the name attribute has the prefix "subjectAltName:". If more than one subject alternative name is present in the certificates, you can add an index prefixed with a pound sign (**#**).

```
d <true|false>
--name-attribute-contains-domain=<true|false>
```

Optional. Indicates whether the specified name attribute contains a fully-qualified account name, with both a username part and a domain name part. The default value is **false**.

config-external-auth

Configures the External Authentication method.

SYNOPSIS

```
config-external-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-e <true|false> | --enabled=<true|false>]
  [-a value | --request-attribute=value] [-r value | --request-header=
  value] [-o value | --request-cookie=value] [-f <true|false> |
  --use-authentication-filter=<true|false>] [-x value | --expression=
  value] [-d <true|false> | --downcase=<true|false>] [-s <true|false> |
  --require-ssl=<true|false>] [-h value | --allowed-hosts=value]
  {-Rvalue}
```

OVERVIEW

Use this command to configure External Authentication. The authentication method can either be used as the main authentication method, as configured by the **set-auth-mode** command, or as a complementary authentication method where it is combined with the main method.

- Typically, this is used as the main method when the clients can access the server(s) only through a proxy or a load-balancer. To use it as the main authentication method, first configure and enable the method using this command and then set it to the main method using the **set-auth-mode** command.
- Typically, this is used as a complementary method when the clients can access the server(s) both directly and through a proxy or a load-balancer. To use it as a complementary method, simply configure and enable the method using this command.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-e <true|false>
--enabled=<true|false>
```

Optional. Specifies whether External Authentication should be enabled. The default value is **true**.

```
-a value
--request-attribute=value
```

Optional. The name of the HTTP request attribute containing the name of the authenticated user. The **--request-attribute**, **--request-header**, **--request-cookie**, and **--use-authentication-filter** arguments are mutually exclusive. The default value is **REMOTE_USER**.

```
-r value
--request-header=value
```

Optional. The name of the HTTP header containing the name of the authenticated user. The **--request-attribute**, **--request-header**, **--request-cookie**, and **--use-authentication-filter** arguments are mutually exclusive.

```
-o value
--request-cookie=value
```

Optional. The name of the HTTP cookie containing the name of the authenticated user. The **--request-attribute**, **--request-header**, **--request-cookie**, and **--use-authentication-filter** arguments are mutually exclusive.

```
-f <true|false>
--use-authentication-filter=<true|false>
```

Optional. Specifies whether the name of the authenticated user is provided by a custom Authentication Filter (as the value of the **getUserPrincipal()** method of **javax.servlet.http.HttpServletRequest**). The **--request-attribute**, **--request-header**, **--request-cookie**, and **--use-authentication-filter** arguments are mutually exclusive. The default value is **false**.

```
-x value
--expression=value
```

Optional. A regular expression used to filter the username extracted from the specified HTTP request attribute. The value of the regular expression's first capturing group is used as the new username. A typical scenario is to extract the username from a composite name containing both username and domain name when using the **collapse domains** option. For example, the regular expression **"\S+\\(\S+)"** can be used to extract the username from a value in the format **domain\username**. Be sure to enclose the specified expression in quotes and to quote all special characters that might otherwise be consumed by the command line shell.

```
-d <true|false>
--downcase=<true|false>
```

Optional. Specifies whether the username should be converted to lowercase. The default value is **false**.

```
-s <true|false>
--require-ssl=<true|false>
```

Optional. Specifies whether a secure HTTPS connection is required to perform External Authentication. The default value is **false**.

```
-h value
--allowed-hosts=value
```

Optional. A comma-separated list of hostnames and/or IP addresses of the client computers that are permitted to perform external authentication. If this or at least one -R argument is not specified, then all client computers are permitted to perform external authentication. Because this is a potential security risk, we strongly recommend to restrict the permissions to use this feature. Typically, this feature is locked so only proxies or load-balancers are permitted to use it.

A scenario where all client computers can be allowed to use this feature is when a custom Post Authentication Filter is also in use. Then this filter would be responsible for performing the final authorization, for instance by validating additional HTTP headers.

```
-Rvalue
```

Optional. A regular expression (in the syntax supported by **java.util.regex.Pattern**) that should match IP addresses of remote hosts that are permitted to perform external authentication. See the **--allowed-hosts** argument. This argument can be specified multiple times with different values.

config-impersonation-auth

Configures the Impersonation authentication method.

SYNOPSIS

```
config-impersonation-auth [-c value | --configuration=value] [-b value
  | --bootstrap-config=value] [-e <true|false> | --enabled=<true|
  false>] [-s <true|false> | --require-ssl=<true|false>] [-p <true|
  false> | --use-post-auth-filter=<true|false>] [-u value |
  --allowed-users=value] [-h value | --allowed-hosts=value] {-Rvalue}
```

OVERVIEW

Use this command to configure the Impersonation authentication method. This is a complementary authentication method that certain users, known as impersonators, can use to assume the identity of a specified user. This feature is typically used by the Spotfire Web Player.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-e <true|false>
--enabled=<true|false>
```

Optional. Specifies whether the Impersonation authentication method should be enabled. The default value is **true**.

```
-s <true|false>
--require-ssl=<true|false>
```

Optional. Specifies whether a secure HTTPS connection is required to use the Impersonation feature. The default value is **false**.

```
-p <true|false>
--use-post-auth-filter=<true|false>
```

Optional. Specifies whether the Impersonation feature should apply the configured Post Authentication Filter after switching the identity. The default value is **true**.

```
-u value
--allowed-users=value
```

Optional. This argument is deprecated and ignored. Add the users to the **Impersonator group** instead.

```
-h value
--allowed-hosts=value
```

Optional. A comma-separated list of hostnames and/or IP addresses of the client computers that are permitted to use the Impersonation feature. If this, or at least one -R argument, is not specified, then all client computers are permitted to use the feature. Because this is a potential security risk, we strongly recommend restricting the permissions to use it. Typically, this feature is locked so that only the computers running the Spotfire Web Player are permitted to use it.

```
-Rvalue
```

Optional. A regular expression (in the syntax supported by **java.util.regex.Pattern**) that should match IP addresses of remote hosts that are permitted to use the Impersonation feature. See also the **--allowed-hosts** argument. This argument can be specified multiple times with different values.

config-import-export-directory

Configures the library import/export directory.

SYNOPSIS

```
config-import-export-directory [-c value | --configuration=value] [-b
  value | --bootstrap-config=value] [-p value | --path=value]
```

OVERVIEW

Use this command to configure the library import/export directory. All library import and export operations are performed from or to this directory. It can be a local directory, or it can reside on a shared disk.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-p value
--path=value
```

Optional. The path to the import/export directory. The default value is **<installation directory>/tomcat/application-data/library**.

config-jmx

Configures the JMX RMI connector.

SYNOPSIS

```
config-jmx [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-e <true|false> | --enabled=<true|false>]
  [-a <true|false> | --authentication-enabled=<true|false>] [-A <true|
  false> | --authorization-enabled=<true|false>] [-s <true|false> |
--ssl-enabled=<true|false>] [-r <true|false> |
--registry-ssl-enabled=<true|false>] [-n <true|false> |
--need-client-auth=<true|false>] [-R value | --registry-port=value]
[-p value | --connector-port=value] [-j value | --jaas-config=value]
```

OVERVIEW

Use this command to configure the JMX RMI connector. This connector can be used for connecting to the Spotfire Server for monitoring and management purposes.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-e <true|false>
--enabled=<true|false>
```

Optional. Specifies whether the RMI connector is enabled. The default value is **false**.

```
-a <true|false>
--authentication-enabled=<true|false>
```

Optional. Specifies whether authentication is enabled for the RMI connector. The default value is **true**.

```
-A <true|false>
--authorization-enabled=<true|false>
```

Optional. Specifies whether authorization is enabled for the RMI connector. Authorization requires authentication to be enabled and works only with the default value of **jaas-config**. The default value is **true**.

```
-s <true|false>
--ssl-enabled=<true|false>
```

Optional. Specifies whether SSL is enabled for the RMI connector. The default value is **false**.

```
-r <true|false>
--registry-ssl-enabled=<true|false>
```

Optional. Specifies whether SSL is enabled for the RMI registry. The default value is **false**.

```
-n <true|false>
--need-client-auth=<true|false>
```

Optional. Specifies whether SSL client authentication is required. The default value is **false**.

```
-R value
--registry-port=value
```

Optional. The port for the RMI registry. The default value is **1099**.

```
-p value
--connector-port=value
```

Optional. The port for the RMI connector. The default value is **1099**.

```
-j value
--jaas-config=value
```

Optional. The JAAS configuration entry to use for authentication. Requires authentication to be enabled. User accounts for the default authentication implementation are created by the **create-jmx-user** command. The default value is **SpotfireJmx**.

config-kerberos-auth

Configures the authentication service used with the Kerberos authentication method.

SYNOPSIS

```
config-kerberos-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-p value | --service-principal-name=value>
  [-k value | --keytab-file=value] [-d <true|false> | --enable-debug=
  <true|false>]
```

OVERVIEW

Use this command to configure the authentication service used with Kerberos authentication method.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-p value
--service-principal-name=value
```

Required. The Kerberos service principal name (SPN) used by the server.

```
-k value
--keytab-file=value
```

Optional. The path to the Kerberos file containing the **keytab** entry for the specified SPN. If the specified path contains any Java system properties (for example as in the default value for this argument), they are automatically expanded. The default value is **${java.home}/lib/security/spotfire.keytab**.

```
-d <true|false>
--enable-debug=<true|false>
```

Optional. Specifies whether extra debug logging should be enabled for the Kerberos authentication service. The default value is **false**.

config-ldap-group-sync

Configures group synchronization for an LDAP configuration.

SYNOPSIS

```
config-ldap-group-sync [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <--id=value> [--group-sync-enabled=<true|
  false>] [--schedules=value] [--clear-schedules] [--group-names=value]
  [--clear-group-names] [--clear-all] [--filter-users-by-groups=<true|
  false>] [--group-search-filter=value] [--group-name-attribute=value]
  [--supports-member-of=<true|false>] [--member-attribute=value]
  [--ignore-member-groups=<true|false>]
```

OVERVIEW

Use this command to configure group synchronization for an LDAP configuration used with the User Directory LDAP provider.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
--id=value
```

Required. Specifies the identifier of the LDAP configuration for which to configure group synchronization.

```
--group-sync-enabled=<true|false>
```

Optional. Specifies whether group synchronization is enabled for this LDAP configuration. The default value is **true**.

```
--schedules=value
```

This argument is deprecated from version 5.0 and is replaced with the similarly named argument for the “create-ldap-config” on page 231 and “update-ldap-config” on page 288 commands, because the synchronization schedules are now used for both user and group synchronization.

Optional. Specifies a comma-separated list of schedules for when the LDAP synchronization is performed. The schedules are given in a cron-compatible format, where each schedule consists of either five fields or one shorthand label.

The five fields are, from left to right, with their valid ranges: minute (0-59), hour (0-23), day of month (1-31), month (1-12) and day of week (0-7, where both 0 and 7 indicate Sunday). A field can be configured with the wildcard character *****, indicating that any moment in time matches this field. A group synchronization is triggered when all fields match the current time. If both day of month and day of week have non-wildcard values, then only one of them must match.

You can use the following shorthand labels instead of the full cron expressions:

@yearly or @annually: run once a year (equivalent to 0 0 1 1 *)

@monthly: run once a month (equivalent to 0 0 1 * *)

@weekly: run once a week (equivalent to 0 0 * * 0)

@daily or @midnight: run once a day (equivalent to 0 0 * * *)

@hourly: run once an hour (equivalent to 0 * * * *)

@minutely: run once a minute (equivalent to * * * * *)

@reboot or @restart: run every time the Spotfire Server is started

Refer to the Wikipedia overview article on the cron scheduler.

`--clear-schedules`

This argument is deprecated from version 5.0 and is replaced with the similarly named argument for the "update-ldap-config" on page 288 command, because the synchronization schedules are now used for both user and group synchronization.

Optional. if you specify this argument, the LDAP synchronization schedules are cleared from the LDAP configuration. This argument can be used together with the **--schedules** argument to remove all old schedules before adding the new.

`--group-names=value`

Optional. Specifies the account names or the distinguished names (DNs) of the groups to be synchronized.

`--clear-group-names`

Optional. If you specify this argument, the list of group names synchronized are cleared from the LDAP configuration. This argument can be used with the **--group-names** argument to remove all old group names before adding the new.

`--clear-all`

Optional. Clears from the LDAP configuration all group synchronization-related configuration options.

As of Spotfire Server 5.0 and later, this option does ***not*** clear the LDAP synchronization schedules.

`--filter-users-by-groups=<true|false>`

Optional. Specifies whether users should be filtered by groups, so that only users that are members of the synchronized groups are synchronized.

`--group-search-filter=value`

Optional, unless the LDAP server type is set to **Custom** using the **--type** parameter. Specifies an LDAP search expression filter to use when searching for groups.

- For Active Directory servers, the parameter value defaults to **objectClass=group**.
- For Sun ONE Directory Servers, it defaults to **&(|(objectclass= nsManagedRoleDefinition)(objectClass=nsNestedRoleDefinition))(objectclass= ldapSubEntry)**.
- For Sun Java System Directory Servers, it defaults to **objectClass= groupOfUniqueNames**.

`--group-name-attribute=value`

Optional, unless the LDAP server type is set to **Custom** using the **--type** parameter. Specifies the name of the LDAP attribute containing the group account names:

- For Active Directory servers, the value defaults to **sAMAccountName**.
- For any version of the Sun directory servers with a default configuration, it defaults to **cn**.

`--supports-member-of=<true|false>`

Optional, unless the LDAP server type is set to **Custom** using the **--type** parameter. Specifies whether the LDAP servers support a **memberOf**-like attribute on the user accounts that contain the names of the groups or roles that the users are members of. In general, this is true for all Microsoft Active Directory servers and all types of Sun directory servers.

`--member-attribute=value`

Optional, unless the LDAP server type is set to **Custom** using the **--type** parameter. For all LDAP servers with support for a **memberOf**-like attribute, this argument specifies the name of the LDAP attribute on the user account that contains the names of the groups or roles that the user is a member of. In general, this includes all Microsoft Active Directory server and all types of Sun Directory Servers.

- For Microsoft Active Directory servers, the parameter value defaults to **memberOf**.
- For Sun ONE Directory Servers, it defaults to **nsRole**.
- For Sun Java System Directory Server version 6.0 or later, it defaults to **isMemberOf**. To use the roles with the Sun Java System Directory Server, override the default value by setting this argument to **nsRole**.

For some LDAP servers with configurations of type **Custom**, there is no **memberOf**-like attribute. In those cases, this argument specifies the LDAP attribute on the group account that contains the names of its members.

All configurations of this type use a far less efficient group synchronization algorithm that generates more traffic to the LDAP servers, because the Spotfire Server first has to search for the distinguished names (DNs) of the group members within the groups, and then perform repeated lookups to translate the member DN to the correct accountname.

`--ignore-member-groups=<true|false>`

Optional, unless the LDAP server type is set to **Custom** using the **--type** parameter. Determines whether the group synchronization mechanism should recursively traverse the synchronized groups' non-synchronized subgroups and include their members in the search result.

For Microsoft Active Directory servers, the parameter value defaults to **false** so all inherited group memberships are correctly reflected. For any version of the Sun Directory Servers, it defaults to **true**, because the role and groups mechanisms in those servers automatically include those members.

config-ldap-userdir

Configures the LDAP User Directory mode.

SYNOPSIS

```
config-ldap-userdir [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-l value | --ldap-configs=value] [-s
  <true|false> | --group-sync-enabled=<true|false>] [-t value |
  --sleep-time=value]
```

OVERVIEW

Use this command to configure the LDAP User Directory mode. If no arguments are specified, the command displays the current configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-l value
--ldap-configs=value
```

Optional. A comma-separated list of LDAP configuration references. All referenced LDAP configurations must already exist. To create a new LDAP configuration, use the **create-ldap-config** command. When specifying more than one reference, make sure to enclose the list of references in quotes.

```
-s <true|false>
--group-sync-enabled=<true|false>
```

Optional. This argument is deprecated and is ignored. Use the **config-ldap-group-sync** command to enable or disable group synchronization for each LDAP configuration instead.

```
-t value
--sleep-time=value
```

Optional. The number of minutes between each synchronization. The sleep time setting is used only for LDAP configuration entries without group synchronization schedules. If an LDAP configuration entry has a synchronization schedule defined, then this value is ignored. The default value is **60**.

config-library-external-data-storage

Configures the external library data storage.

SYNOPSIS

```
config-library-external-data-storage [-c value | --configuration=value]
  [-b value | --bootstrap-config=value] [-t value | --tool-password=
  value] <-e <true|false> | --enabled=<true|false>> [-s value |
  --external-storage=value] [-f | --force]
```

OVERVIEW

Use this command for general configuration of the external library data storage.

When this feature is enabled, the structure of the library is stored in the TIBCO Spotfire Server database, while the actual data of library items are stored elsewhere.

The library must be empty when you switch to or from an external data storage. The prescribed procedure for switching is to export the entire library, empty the library, change the configuration, and then import the library. Switching storage modes with items in the library causes data to be lost.

When you change the external library data storage configuration with this command, a query is made to the TIBCO Spotfire Server database to make sure that the library is empty. This check can be overridden by using the **--force** argument.

Presently, Spotfire supports two options for external data storage: storing on the server's file system, or storing on Amazon S3. After you enable this feature, you must configure the storage using the **config-external-library-file-storage** or **config-external-library-s3-storage** commands.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-e <true|false>
--enabled=<true|false>
```

Required. Specifies whether external library data storage should be enabled.

```
-s value
--external-storage=value
```

Optional. The external storage to use. The following names are valid: **FILE_SYSTEM** and **AMAZON_S3**.

```
-f
--force
```

Optional. Indicates that the tool should change the library configuration even if the library is not empty.

config-library-external-file-storage

Configures the file system storage of library item data.

SYNOPSIS

```
config-library-external-file-storage [-c value | --configuration=value]
  [-b value | --bootstrap-config=value] <-p value | --path=value>
```

OVERVIEW

Use this command for configuring file system storage of library data.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-p value
--path=value
```

Required. The path to the directory where library data is stored. Supply the value **DEFAULT** to use the TIBCO Spotfire Server's default location for storing library data on file system.

config-library-external-s3-storage

Configures the Amazon S3 storage of library item data.

SYNOPSIS

```
config-library-external-s3-storage [-c value | --configuration=value]
  [-b value | --bootstrap-config=value] [--bucket-name=value]
  [--access-key=value] [--secret-key=value] [--region=value]
  [--threads=value] [--chunk-size=value] [--threshold=value]
```

OVERVIEW

Use this command for configuring the Amazon S3 storage of library data.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
--bucket-name=value
```

Optional. The Amazon S3 bucket where library data is stored.

```
--access-key=value
```

Optional. The access key for connecting to Amazon S3.

```
--secret-key=value
```

Optional. The secret key for connecting to Amazon S3.

```
--region=value
```

Optional. The Amazon S3 region to which to connect. If not configured explicitly, server uses the default region.

```
--threads=value
```

Optional. The maximum number of threads used for uploading to Amazon S3.

```
--chunk-size=value
```

Optional. The maximum number of bytes in a chunk when the data is chunked before transfer to Amazon S3.

```
--threshold=value
```

Optional. The number of bytes above which the transferred data is split into configurable-sized chunks, and then transferred separately to Amazon S3.

config-login-dialog

Configures the client log in dialog behavior.

SYNOPSIS

```
config-login-dialog [-c value | --configuration=value]
  [-b value | --bootstrap-config=value]
  [-s value | --show-login-dialog=value] [-o <true|false> |
  --allow-work-offline=<true|false>] [-d value |
  --offline-days-permitted=value] [-r <true|false> |
  --allow-remember-me=<true|false>] [-u <true|false> |
  --allow-user-provided-credentials=<true|false>] [-R value | --rss=
  value]
```

OVERVIEW

Use this command to configure the behavior of the client log in dialog.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-s value
--show-login-dialog=value
```

Optional. Controls whether the log in dialog should be displayed. Valid values are:

- **always**: Show the dialog even if the user chose Save my login information
- **never**: Never show the dialog.
  Use this option only with one of the single sign-on methods: NTLM, Kerberos, and X.509 Client Certificates.
- **standard**: Show the dialog only if the user did not chose Save my login information.

The default value is **standard**.

```
-o <true|false>
--allow-work-offline=<true|false>
```

Optional. Controls whether users should be allowed to work offline or if they must always log in. The default value is **true**.

```
-d value
--offline-days-permitted=value
```

Optional. Controls how long users can choose to work offline before they are forced to log in. Setting the value to **-1** means that users are never forced to connect to the Spotfire Server. The default value is **-1**.

```
-r <true|false>
--allow-remember-me=<true|false>
```

Optional. Controls whether a user can select to store the log in information for future automatic login, or if he or she must always provide user name and password when logging in. The default value is **true**.

```
-u <true|false>
--allow-user-provided-credentials=<true|false>
```

Optional. Controls whether users should be able to enter their own credentials in the log in dialog. The default value is **true**.

```
-R value
--rss=value
```

Optional. The URL to an RSS feed to be shown in the log in dialog. The feed must be RSS 2.0 compliant. that HTML in the RSS feed is not supported.

config-ntlm-auth

Configures the authentication service used with the NTLM authentication method.

SYNOPSIS

```
config-ntlm-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-S value | --server=value] [-d value |
  --domain-name=value] [-D value | --domain-controller=value] [-a value
| --account-name=value] [-p value | --password=value] [-n value |
--dns-servers=value] [-s value | --ad-site=value] [-t value |
--dns-cache-ttl=value] [-l value | --localhost-netbios-name=value]
[-i value | --connection-id-header-name=value]
```

OVERVIEW

Use this command to configure the authentication service used with NTLM authentication method.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-S value
--server=value
```

Optional. The name of the cluster server to which the specified configuration parameters should be applied. If no name is specified, the parameters apply to all servers in the cluster. It is typically used to add a server-specific account name or localhost NetBIOS name (see **--account-name** and **--localhost-netbios-name** options).

```
-d value
--domain-name=value
```

Required, unless the **--domain-controller** argument is specified, or if the **--server** argument is specified and this parameter is already specified for the global configuration.

The DNS name of the Windows domain. The specified domain name automatically resolves into domain controller hostnames. It is also possible to use the **--domain-controller** argument to specify a domain controller hostname directly. The **--domain-name** and **--domain-controller** arguments are mutually exclusive.

```
-D value
--domain-controller=value
```

Required, unless the **--domain-controller** argument is specified, or if the **--server** argument is specified and this parameter is already specified for the global configuration.

The DNS hostname of an Active Directory domain controller. It is also possible to use the **--domain-name** argument to specify a domain name that automatically resolves to domain controller hostnames. The **--domain-name** and **--domain-controller** arguments are mutually exclusive.

```
-a value
--account-name=value
```

Required, unless the **--server** argument is specified and this parameter is already specified for the global configuration.

Specifies the fully qualified name of the Active Directory computer account to be used by the NTLM authentication service. This account must be a proper computer account created solely for the purpose of running the NTLM authentication service. It can neither be an ordinary user account, nor an account of an existing computer. that the name of an Active Directory computer account always contains a dollar-sign; for example, **ntlm-svc$@research.example.com**. The local part of the account name (excluding the dollar-sign) must not exceed 15 characters. Also, because of the dollar-sign, always make sure to enclose this parameter value in quotes and possibly also escape the dollar-sign.

If more than one server is in the cluster, each server must use its own account. Sometimes it is possible to share the computer account. The extra servers must then be configured with server-specific localhost NetBIOS names (see the **--localhost-netbios-name** option).

```
-p value
--password=value
```

Required, unless the **--server** argument is specified and this parameter is already specified for the global configuration.

Specifies the password for the computer account that is to be used by the NTLM authentication service.

```
-n value
--dns-servers=value
```

Optional. A comma-separated list of IP addresses for the DNS servers associated with the Windows domain. When no DNS servers are specified, the NTLM authentication service falls back to the server computer default DNS server configuration.

```
-s value
--ad-site=value
```

Optional. The Active Directory Site where the Spotfire system is located. Specifying an Active Directory Site can potentially improve performance, because the NTLM authentication service then communicates only with the local Domain Controllers.

```
-t value
--dns-cache-ttl=value
```

Optional. The length of time (in milliseconds) name server lookups should be cached. The default value is **5000** ms.

```
-l value
--localhost-netbios-name=value
```

Optional. The NetBIOS name used by each server to identify its connection to the domain controller. The default value is derived from the account name specified by the **--account-name** option.

Because the domain controller allows only one connection per NetBIOS name, each additional server after the first must use this option with the **--server** argument to specify a unique localhost NetBIOS name, not exceeding 15 characters in length. The parameter is necessary only to specify when more than one server is in the cluster.

```
-i value
--connection-id-header-name=value
```

Optional. The name of an HTTP header containing unique connection IDs in environments where the server is located behind a proxy or load-balancer that does not properly provide the server with the client IP address.

The specified HTTP header must contain unique connection IDs for each client connection and is thus typically based on the client IP address and the connection port number on the client side.

EXAMPLES

Configuring the NTLM authentication service for the **research.example.com** Windows domain:

```
config.bat config-ntlm-auth --domain-name research.example.com --account-name
"ntlm-svc\$@research.example.com" --password 53cr3t
```

Configuring the NTLM authentication service for using the Active Directory Domain Controller dc.research.example.com:

```
config-ntlm-auth --domain-controller dc.research.example.com --account-name "ntlm-svc\
$@research.example.com" --password 53cr3t
```

Configuring the NTLM authentication service for the Active Directory Site VIENNA within the **research.example.com** Windows domain:

```
config-ntlm-auth --domain-name research.example.com --account-name "ntlm-svc\$@
research.example.com" --password 53cr3t --ad-site=VIENNA
```

config-post-auth-filter

Configures the Post Authentication Filter.

SYNOPSIS

```
config-post-auth-filter [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-f value | --filter-class=value] [-s value
  | --filter-config=value] [-d value | --default-filter-config=value]
```

OVERVIEW

Use this command to configure the Post Authentication Filter. If no argument is provided, the command simply lists the current configuration and exits.

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-f value
--filter-class=value
```

Optional. The fully-qualified name of the class implementing the **com.spotfire.server.security.PostAuthenticationFilter** API. If the argument is **none**, the current value of this configuration option is cleared.

```
 -s value
--filter-config=value
```

Optional. The filter configuration. The semantics of the configuration argument is specific to the actual filter implementation. For example, it could be a configuration name, a filename, or a list of key/value pairs. If the argument is **none**, the current value of this configuration option is cleared.

```
-d value
--default-filter-config=value
```

Optional. The configuration for the default filter that is always in place. Valid arguments are **block** and **autocreate**. See the default filter section for more information.

THE DEFAULT FILTER IMPLEMENTATION

You can use the default implementation of the Post Authentication Filter for access control if you are using an external authentication source, such as LDAP or Windows NT Domain, in combination with the Database User Directory mode. If you are using a different combination of Authentication and User Directory, the filter has no effect.

The default implementation has two modes:

- The user is allowed access only if he or she already exists in the user directory. (To configure this option, use **--default-filter-config=block**.)
- The user is allowed access regardless whether or not he or she exists in the user directory. He or she is added to the User Directory. (To configure this use **--default-filter-config=autocreate**.)

EXAMPLES

Configuring the default filter to block users not in the User Directory (the default behavior):

config-post-auth-filter --filter-class=
com.spotfire.server.security.PostAuthenticationFilterImpl --filter-config=block

Configuring the default filter to automatically create users not in the User Directory:

config-post-auth-filter --default-filter-config=autocreate

Configuring a custom filter implementation with the configuration in a file:

config-post-auth-filter --filter-class=com.example.MyPostAuthenticationFilter
--filter-config="c:\tibco\tss\6.5\tomcat\webapps\spotfire\WEB-INF\
paf_config.properties"

config-two-factor-auth

Configures two-factor authentication.

SYNOPSIS

```
config-two-factor-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-e <true|false> | --enabled=<true|false>]
```

OVERVIEW

Use this command to configure two-factor authentication. If no argument is provided, the command simply lists the current configuration and exits.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-e <true|false>
--enabled=<true|false>
```

Optional. Specifies whether two-factor authentication should be enabled.

TWO-FACTOR AUTHENTICATION

You can enable two-factor authentication on the Spotfire Server by combining the primary authentication method (typically the **BASIC** authentication method) with the **CLIENT_CERT** authentication method. With two-factor authentication enabled, the server requires the name of the authenticated user to match the username in the provided X.509 certificate.

To set up two-factor authentication, follow these steps:

1 Set and configure the primary authentication method.

2 Configure the **CLIENT_CERT** authentication method.

3 Enable two-factor authentication.

config-userdir

Configures the User Directory.

SYNOPSIS

```
config-userdir [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-m value | --mode=value] [-C <true|false>
  | --collapse-domains=<true|false>] [-S <true|false> |
  --safe-synchronization=<true|false>] [-s value | --domain-name-style=
  value] [-u <true|false> | --unsafe-domain-name-style-allowed=<true|
  false>]
```

OVERVIEW

Use this command to configure the User Directory.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-m value
--mode=value
```

Optional. The name of the User Directory mode to use. Supported values are **database**, **ldap**, and **windows**. The default value is **database**. The current value does not change unless the argument is specified explicitly.

```
-C value
--collapse-domains=value
```

Optional. Indicates whether external domains should be collapsed into the internal **SPOTFIRE** domain, which is the domain used when running the User Directory in database mode. The default value is **false**. The current value does not change unless the argument is specified explicitly.

All users belong to the same domain when this feature enabled. If multiple users with the same account name from different external domains exist, they now share a Spotfire account. Because this could pose a security problem, this feature should be used with care.

```
-S <true|false>
--safe-synchronization=<true|false>
```

Optional. If set to **true**, the User Directory does not disable users that it cannot find during LDAP or Windows NT synchronization. This argument has no effect if the User Directory is running in Database mode. The default value is **false**. The current value does not change unless the argument is specified explicitly.

```
-s value
--domain-name-style=value
```

Optional. The domain name style used by the server. Supported values are **dns** and **netbios**. The default value is **dns**. The current value does not change unless the argument is specified explicitly.

```
-u <true|false>
--unsafe-domain-name-style-allowed=<true|false>
```

Optional. If set to **true**, the server allows incompatible domain name style settings, instead of refusing to start. This option should be used with care. It can lead to many of users and groups being imported to the User Directory with invalid domain names. The default value is **false**. The current value does not change unless the argument is specified explicitly.

config-windows-userdir

Configures the Windows User Directory mode.

SYNOPSIS

```
config-windows-userdir [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-d value | --domains=value] [-t value |
  --sleep-time=value] [--schedules=value]
```

OVERVIEW

Use this command to configure the Windows User Directory mode. If no arguments are specified, the command displays the current configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-d value
--domains=value
```

Optional. A comma-separated list of domain names. When specifying more than one domain name, make sure to enclose the list of names in quotes.

```
-t value
--sleep-time=value
```

Optional. The number of minutes between each synchronization. The **--sleep-time** and **--schedules** arguments are mutually exclusive. If neither the **--sleep-time** argument nor the **--schedules** argument is specified, the synchronization is performed with a sleep time of 60 minutes.

```
--schedules=value
```

Optional. A comma-separated list of schedules for when the synchronization should be performed. The **--sleep-time** and **--schedules** arguments are mutually exclusive. The schedules are given in a cron-compatible format, where each schedule consists of either five fields or one shorthand label. Make sure to enclose the value in double quotes.

The five fields are, from left to right, with their valid ranges: minute (0-59), hour (0-23), day of month (1-31), month (1-12) and day of week (0-7, where both 0 and 7 indicate Sunday). You can configure a field with the wildcard character *****, indicating that any moment in time matches this field. An LDAP synchronization is triggered when all fields match the current time. If both day of month and day of week have non-wildcard values, then only one of them has to match.

You can use the following shorthand labels instead of the full cron expressions:

@yearly or @annually: run once a year (equivalent to 0 0 1 1 *)

@monthly: run once a month (equivalent to 0 0 1 * *)

@weekly: run once a week (equivalent to 0 0 * * 0)

@daily or @midnight: run once a day (equivalent to 0 0 * * *)

@hourly: run once an hour (equivalent to 0 * * * *)

@minutely: run once a minute (equivalent to * * * * *)

@reboot or @restart: run every time the Spotfire Server is started

Consult the Wikipedia article for an overview of the cron scheduler:

http://en.wikipedia.org/wiki/Cron

create-default-config

Creates a new server configuration file containing the default configuration.

SYNOPSIS

```
create-default-config [-f | --force] [export file]
```

OVERVIEW

Use this command to export a default server configuration to a file. The configuration in the file can be edited and then imported into the server database using the **import-config** command.

OPTIONS

```
-f
--force
```

Optional. Indicates that the tool should overwrite an existing destination file.

```
[export file]
```

Optional. The path to the configuration file to create. The default value is **configuration.xml**.

create-jmx-user

Creates a new JMX user account.

SYNOPSIS

```
create-jmx-user [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-u value | --username=value> [-p value |
  --password=value] [-l value | --access-level=value]
```

OVERVIEW

Use this command to create a new JMX user account. The account can be used only to access status information for the server through the JMX protocol. It cannot be used by users logging in to the server using a Spotfire client or an HTML browser.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-u value
--username=value
```

Required. The name of the JMX user to create.

```
-p value
--password=value
```

Optional. The new JMX user password.

```
-l value
--access-level=value
```

Optional. The access level for the new user. Can be either **r** or **rw**. A user with the **rw** access level can read and modify any writable attributes. The default value is **r**.

create-join-db

Configures the default join database.

SYNOPSIS

```
create-join-db [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-t value | --type=value> <-d value |
  --database-url=value> <-u value | --username=value> [-p value |
  --password=value] [-i value | --min-connections=value] [-a value |
  --max-connections=value] [-v | --validate]
```

OVERVIEW

Use this command to configure the default join database.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

`-t value`
`--type=value`

Required. The database type and the driver to use. Must match the type name of one of the enabled data source templates.

`-d value`
`--database-url=value`

Required. The JDBC URL to the database. Because this argument usually contains special characters, be sure to escape those characters or enclose the values in quotes.

`-u value`
`--username=value`

Required. The database account username.

`-p value`
`--password=value`

Optional. The database account password.

`-i value`
`--min-connections=value`

Optional. The minimum number of connections to keep in the connection pool. The default value is **0**.

`-a value`
`--max-connections=value`

Optional. The maximum number of connections to keep in the connection pool. The default value is **0**.

`-v`
`--validate`

Optional. Indicates whether the created configuration should be validated by attempting to connect to the database using the specified connection information.

create-ldap-config

Creates a new LDAP configuration for authentication and/or the User Directory LDAP provider.

SYNOPSIS

```
create-ldap-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <--id=value> [--discover] [-t value |
  --type=value] [-s value | --servers=value] [-n value |
  --context-names=value] [-u value | --username=value] [-p value |
  --password=value] [--schedules=value] [--user-search-filter=value]
  [--user-name-attribute=value] [--authentication-attribute=value]
  [--security-authentication=value] [--referral-mode=value]
  [--request-control=value] [--page-size=value] [--import-limit=value]
  [--user-display-name-attribute=value]
  [--group-display-name-attribute=value] {-Ckey=value}
```

OVERVIEW

Use this command to create a new LDAP configuration for authentication and/or user directory mode.

OPTIONS

`-c value`
`--configuration=value`

Optional. The path to the server configuration file. The default value is **configuration.xml**.

`-b value`
`--bootstrap-config=value`

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

`--id=value`

Required. Specifies the identifier for the LDAP configuration to create.

`--discover`

Optional. Specifies whether to attempt to automatically create an LDAP configuration based on the information available from the DNS service. The discover mode works only when the desired LDAP server has registered SRV records in the DNS service used by the computer where this command is being invoked. This is typically the case for Active Directory LDAP servers. This argument is mutually exclusive with the **-t/--type**, **-s/--servers** and **-n/--context-names** arguments.

`-t value`
`--type=value`

Required, unless the **--discover** option is used. The type of LDAP server. The following names are valid types:

- **ActiveDirectory**
- **SunOne**
- **SunJavaSystem**
- **Custom**

If you specify any of the first three types, a type-specific configuration template is automatically applied in runtime, so that the most fundamental configuration options are automatically configured.

If you specify a **Custom** LDAP server type, there is no such configuration template, and you must specify explicitly all the configuration options. When you use a custom LDAP configuration for authentication or with the User Directory LDAP provider, you must specify the argument **--user-search-filter** and **--user-name-attribute**. If you use such an LDAP configuration for group synchronization, you must also specify additional parameters when running the **config-ldap-group-sync** command. See “config-ldap-group-sync” on page 214.

`-s value`
`--servers=value`

Required, unless the **--discover** option is used. A whitespace-separated list of LDAP server URLs. An LDAP server URL has the format **<protocol>://<server>[:<port>]**:

- **<protocol>**: Either **LDAP** or **LDAPS**
- **<server>**: The fully qualified DNS name of the LDAP server.

- **<port>**: Optional. Indicates the port number the LDAP service is listening on. The LDAP protocol port number defaults to **389**. The LDAPS protocol port number defaults to **636**. Active Directory LDAP servers also provide a Global Catalog containing forest-wide information, instead of domain-wide information only. By default, the Global Catalog LDAP service listens on port number **3268** (LDAP) or **3269** (LDAPS).

The Spotfire Server does not expect search base, scope, filter, or other additional parameters after the port number in the LDAP server URLs. Such properties are specified using other configuration options for this command.

Examples: LDAP server URLs
LDAP://myserver.example.com
LDAPS://myserver.example.com
LDAP://myserver.example.com:389
LDAPS://myserver.example.com:636
LDAP://myserver.example.com:3268
LDAPS://myserver.example.com:3269

```
-n value
--context-names=value
```

Required, unless the **--discover** option is used. A list of distinguished names (DNs) of the containers holding the LDAP accounts to be visible within the Spotfire Server. When you specify more than one DN, you must separate the DNs using pipe-characters (**|**). If the specified containers contain a large number of users, of which only a few should be visible in the Spotfire Server, you can specify a custom user search filter to include only the designated users (see the **--user-search-filter** argument).

Examples:

CN=users,DC=example,DC=com

OU=project-x,DC=research,DC=example,DC=com

```
-u value
--username=value
```

Required. The name of the LDAP service account to use when searching for users (and optionally also groups) in the LDAP server. This service account does not need to have write permissions, but it must have read permissions for all configured context names (LDAP containers). For most LDAP servers, the account name is the account's distinguished name (DN). For Active Directory, the account name can also be specified in the forms **ntdomain\name** and **name@dnsdomain**.

Examples:

CN=spotsvc,OU=services,DC=research,DC=example,dc=COM

RESEARCH\spotsvc (note: Active Directory only)

spotsvc@research.example.com (note: Active Directory only)

```
-p value
--password=value
```

Optional. The password for the LDAP service account.

```
--schedules=value
```

Optional. A comma-separated list of schedules for when the LDAP synchronization should be performed. The schedules are given in a cron-compatible format, where each schedule consists of either five fields or one shorthand label. Make sure you enclose the value in double quotes. The default value is **@daily**, **@restart**.

The five fields are, from left to right, with their valid ranges: minute (0-59), hour (0-23), day of month (1-31), month (1-12) and day of week (0-7, where both 0 and 7 indicate Sunday). You can also configure a field with the wildcard character *****, indicating that any moment in time matches this field. A group synchronization is triggered when all fields match the current time. If both day of month and day of week have non-wildcard values, then only one of them has to match.

You can also use following shorthand labels instead of the full cron expressions:

@yearly or @annually: run once a year (equivalent to 0 0 1 1 *)

@monthly: run once a month (equivalent to 0 0 1 * *)

@weekly: run once a week (equivalent to 0 0 * * 0)

@daily or @midnight: run once a day (equivalent to 0 0 * * *)

@hourly: run once an hour (equivalent to 0 * * * *)

@minutely: run once a minute (equivalent to * * * * *)

@reboot or @restart: run every time the Spotfire Server is started

Refer to the Wikipedia overview article on the cron scheduler.

```
--user-search-filter=value
```

Usually optional; required for custom LDAP configurations, either when running this command or the **update-ldap-config** command.

Specifies an LDAP search expression filter to use when searching for users.

- For Active Directory servers, the parameter value defaults to **objectClass=user**.
- For any version of the Sun Directory Servers, it defaults to **objectClass=person**.

If you need to identify a subset of users in the specified LDAP containers who should be allowed access to the Spotfire Server, you can specify a more detailed user search filter. For example, the search expression can be expanded so that it also puts restrictions on which groups the users belong to, or which roles they have.

- For Active Directory servers, access can be restricted to only those users belonging to a certain group by using a search expression with the pattern **&(objectClass=user)(memberOf=<groupDN>)** where **<groupDN>** is replaced by the real DN of the group to which the users must belong. If the users are divided among multiple groups, use the pattern **&(objectClass=user)(|(memberOf=<firstDN>)(memberOf=<secondDN>))**. Add extra **(memberOf=<groupDN>)** sub-expressions as needed.

  Active Directory Example:
  &(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)

- For a Sun Java System Directory Server version 6 and later, you can achieve the same effect by using a search expression with the pattern **&(objectClass=person)(isMemberOf=<groupDN>)**. If the users are divided among multiple groups, use the pattern **&(objectClass=person)(|(isMemberOf=<firstDN>)(isMemberOf=<secondDN>))**. Add extra **(isMemberOf=<groupDN>)** sub-expressions as needed.

Sun Java System Directory Server Example:
&(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)

- For Sun ONE Directory Servers and newer Sun Java System Directory Servers or the older iPlanet Directory Server, you can restrict access to only those users having certain specific roles. The search expression for role filtering must match the pattern **&(objectClass=person)(nsRole=<roleDN>)**. If multiple roles are of interest, use the pattern **&(objectClass=person)(|(nsRole=<firstDN>)(nsRole=<secondDN>)**. Add extra **(nsRole=<roleDN>)** sub-expressions as needed.

  Sun ONE Directory Servers Example:
  &(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)

The syntax of LDAP search expression filters is specified by the RFC 4515 document. Consult this documentation for information about more advanced filters.

`--user-name-attribute=value`

Optional, unless the LDAP server type is set to **Custom** using the **--type** parameter. Specifies the name of the LDAP attribute containing the user account names.

- For Active Director servers, the value defaults to **sAMAccountName**.
- For a Sun Java System Directory Server or any older Sun ONE Directory Server or iPlanet Directory Server with a default configuration, it defaults to **uid**.

`--authentication-attribute=value`

Optional; use only for advanced setups. It is not set by default.

Specifies the name of the LDAP attribute containing a user identity that can be used for binding (authenticating) to the LDAP server. This attribute fills no purpose in most common LDAP configurations, but it can be useful in more advanced setups where the distinguished name (DN) does not work for authentication, or where users should be able to log in using a username that does not map directly to an actual LDAP account.

- If you set up SASL with DIGEST-MD5 in an Active Directory environment, the DN does not work for authentication, and the **userPrincipalName** attribute must be used instead. The **--authentication-attribute** argument should then be set to **userPrincipalName** and the **--user-name-attribute** argument should be set to **sAMAccountName**. (The latter value is the default value for an Active Directory LDAP configuration, so there is no need to set it explicitly.) See also the **--security-authentication** argument.
- When you set up SASL with GSSAPI in an Active Directory environment, the DN does not work for authentication and the **sAMAccountName** or **userPrincipalName** attribute must be used instead. The **--authentication-attribute** argument should be set to **sAMAccountName** or **userPrincipalName**, and the **--user-name-attribute** argument should be set to **'sAMAccountName'**. (The latter value is the default value for an Active Directory LDAP configuration, so there is no need to set it explicitly.) See also the **--security-authentication** argument.

Example:

If you set the **--user-name-attribute** argument to **cn** and the **--authentication-attribute** argument to **userPrincipalName** in an Active Directory environment, the users can log in to the Spotfire Server using their CN attribute values, but underneath the hood, the Spotfire Server actually uses the **userPrincipalName** attribute value of the LDAP account with the matching CN for the actual authentication.

`--security-authentication=value`

Optional; use only in advanced setups. The default value is **simple**.

Specifies the security level to use when binding to the LDAP server:

- To enable anonymous binding, it should be set to **none**.
- To enable plain username/password authentication, it should be set to **simple**.
- To enable SASL authentication, it should be set to the name of the SASL mechanism to be used, for instance **DIGEST-MD5** or **GSSAPI**. Use multiple **-C** arguments to set the additional JNDI environment properties that the SASL authentication mechanism typically requires.

If you set up SASL with DIGEST-MD5 in an Active Directory environment, all accounts must use reversible encryption for their passwords. This is typically not the default setting for the domain controller. The **--authentication-attribute** argument must also be used to specify the **userPrincipalName** attribute for the actual authentication to work correctly.

If you set up SASL with GSSAPI in an Active Directory environment, the **--authentication-attribute** argument must be used to specify either the **sAMAccountName** or the **userPrincipalName** attribute and the custom property **kerberos.login.context.name** must be mapped to the JAAS application configuration **SpotfireGSSAPI**. This, is turn, requires a fully working Kerberos configuration file at **<installation directory>/jdk/jre/lib/security/krb5.conf**.

`--referral-mode=value`

Optional. Specifies how LDAP referrals should be handled. Valid arguments are as follows:

- **follow** (automatically follow any referrals). Recommended; the default.
- **ignore** (ignore referrals)
- **throw** (fail with an error).

`--request-control=value`

Optional. The default value is **probe**.

Determines the type of LDAP controls to be used for executing search queries to the LDAP server. The default behavior is to probe the LDAP server for the best supported request control. The paged results control is always preferred, because it provides the most efficient way of retrieving the query result set. You can use the virtual list view control for the same purpose if the paged results control is not supported. The virtual list view control is used automatically, together with a sort control. Both the paged results control and the virtual list view control support a configurable page size, set by the **--page-size** argument.

- To explicitly configure the server for probing, set the argument value to **probe**.
- To configure the server for the paged results control, set the argument value to **PagedResultsControl**.
- To request the virtual list view control, set the argument value to **VirtualListViewControl**.
- To completely disable request controls by setting the argument value to **none**.

`--page-size=value`

Optional. Specifies the page size to be used with the paged results control or the virtual list view control when performing search queries to the LDAP server. The page size value defaults to **2000** for both the paged results control and the virtual list view control.

`--import-limit=value`

Optional. Specifies a threshold that limits the number of users that can be imported from an LDAP server to the Spotfire Server in one query. This can be used to prevent accidentally flooding the Spotfire Server User Directory when you integrate with an LDAP server with tens or even hundreds of thousands of users. By setting an import limit, you can be sure that an unexpected high number of users does not affect server performance. By default, there is no import limit. To request unlimited import explicitly, set the parameter value to **-1**. All positive numbers are treated as an import limit. We recommend for most cases that you leave this parameter untouched.

`--user-display-name-attribute=value`

Optional. Specifies the name of the LDAP attribute containing the user display names.

`--group-display-name-attribute=value`

Optional. Specifies the name of the LDAP attribute containing the group display names.

`-Ckey=value`

Optional; can be specified multiple times with different keys. Specifies additional JNDI environment properties to use when connecting to the LDAP server.

Example: The equivalent of specifying the **--security-authentication=DIGEST-MD5** argument is **-Cjava.naming.security.authentication=DIGEST-MD5**.

EXAMPLES

Create an LDAP configuration for Active Directory:

```
create-ldap-config  --id="ldap1"  --type="ActiveDirectory"  --servers="ldap://
dc01.research.example.com:3268  ldap://dc02.research.example.com:3268"
--context-names="OU=project-x,DC=research,DC=example,DC=com|OU=phbs,DC=
management,DC=example,DC=com"  --username="ldapadmin@research.example.com"
--password="s3cr3t" --schedules="@daily"
```

Create an LDAP configuration for SunONE:

```
create-ldap-config  --id="ldap1"  --type="SunONE"  --servers="ldap://
directory.research.example.com:389"  --context-names="OU=project-x,DC=research,DC=
example,DC=com|OU=phbs,DC=management,DC=example,DC=com"  --username=
"ldapadmin"  --password="s3cr3t" --schedules="@daily"
```

Create an LDAP configuration for Sun Java System Directory:

```
create-ldap-config  --id="ldap1"  --type="SunJavaSystem"  --servers="ldaps://
directory.research.example.com:636"  --context-names="OU=project-x,DC=research,DC=
example,DC=com|OU=phbs,DC=management,DC=example,DC=com"  --username=
"ldapadmin"  --password="s3cr3t" --schedules="@daily"
```

Create an LDAP configuration for a custom LDAP server:

create-ldap-config --id="ldap1" --type="Custom" --servers="ldap://directory.research.example.com"--context-names="OU=project-x,DC=research,DC=example,DC=com|OU=phbs,DC=management,DC=example,DC=com" --user-name-attribute="cn" --search-filter="&(objectClass=person)(isMemberOf=cn=projectX,dc=example,dc=com)" --username="ldapadmin" --password="s3cr3t" --schedules="@daily"

Create an LDAP configuration using the discover mode:

create-ldap-config --id="ldap1" --discover --username="ldapadmin@research.example.com" --password="s3cr3t" --schedules="@daily"

create-user

Creates a new user account.

SYNOPSIS

```
create-user [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-u value | --username=value> [-p value |
  --password=value] [-d value | --display-name=value] [-e value |
  --email=value]
```

OVERVIEW

Use this command to create a new user account. This user can then be promoted to administrator using the **promote-admin** command.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-d value
--display-name=value
```

Optional. The new user's display name.

```
-e value
--email=value
```

Optional. The new user's email address.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-u value
--username=value
```

Required. The name of the new user.

```
-p value
--password=value
```

Optional. The new user’s password.

delete-disabled-users

Deletes disabled users.

SYNOPSIS

```
delete-disabled-users [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-a <true|false> | --keep-once-active-users=
  <true|false>] [-m <true|false> | --keep-group-members=<true|false>]
  [-p <true|false> | --keep-users-with-library-permissions=<true|
  false>] [-l <true|false> | --keep-library-authors=<true|false>] [-f |
  --force]
```

OVERVIEW

Use this command to delete disabled users from the User Directory.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-a <true|false>
--keep-once-active-users=<true|false>
```

Optional. Indicates whether all users that have logged in at least once should be kept. The default value is **'true'**.

```
-m <true|false>
--keep-group-members=<true|false>
```

Optional. Indicates whether all users that are members of at least one group should be kept. The default value is **'true'**.

```
-p <true|false>
--keep-users-with-library-permissions=<true|false>
```

Optional. Indicates whether all users that have explicit library permissions should be kept. The default value is **'true'**.

```
-l <true|false>
--keep-library-authors=<true|false>
```

Optional. Indicates whether all users that have created or modified any Library item should be kept. The default value is **'true'**.

```
-f
--force
```

Optional. Indicates that users should be deleted without need for further confirmation.

delete-disconnected-groups

Deletes disconnected groups.

SYNOPSIS

```
delete-disconnected-groups [-b value | --bootstrap-config=value] [-t
  value | --tool-password=value] [-f | --force]
```

OVERVIEW

Use this command to delete disconnected groups that have been previously synchronized from an LDAP directory from the User Directory.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-f
--force
```

Optional. Indicates that the groups should be deleted without need for further confirmation.

delete-jmx-user

Deletes a JMX user.

SYNOPSIS

```
delete-jmx-user [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-u value | --username=value>
```

OVERVIEW

Use this command to delete a user who can access the server through JMX.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-u value
--username=value
```

Required. The name of the user to be deleted.

delete-library-content

Deletes library content.

SYNOPSIS

```
delete-library-content [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-t value | --tool-password=value] <-i
  value | --items=value> [-d | --database] [-e | --external]
```

OVERVIEW

Use this command to delete library items from the Spotfire database or from external storage on Amazon S3.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the **bootstrap.xml** file. If the tool password is omitted, the command prompts the user for it on the console. Refer to “bootstrap.xml” on page 176.

```
-i value
--items=value
```

Required. A comma-separated list of items (GUIDs) to delete.

```
-d
--database
```

Optional. Deletes entries in the Spotfire library database.

```
-e
--external
```

Optional. Deletes entries in external storage.

delete-user

Deletes a user account.

SYNOPSIS

```
delete-user [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-u value | --username=value>
```

OVERVIEW

Use this command to delete a user account.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-u value
--username=value
```

Required. The name of the user to be deleted.

demote-admin

Revokes full administrator privileges from a user.

SYNOPSIS

```
demote-admin [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-u value | --username=value>
```

OVERVIEW

Use this command to revoke administrator privileges for a user by removing the user account from the Administrator group.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-u value
--username=value
```

Required. The name of the user for which to revoke the administrator privileges. Unless the user is part of the configured default domain, the name of the user needs to include the user's domain name, for instance **DOMAIN\user** or **user@domain**.

enable-user

Enables or disables a user in the Spotfire Database.

SYNOPSIS

```
enable-user [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-u value | --username=value] [-a | --all] [-e
  <true|false> | --enabled=<true|false>]
```

OVERVIEW

Use this command to enable or disable a user in the Spotfire Database. A disabled user does not have access to the Spotfire Server.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-u value
--username=value
```

Optional. The user that should be enabled or disabled. Should not be specified if the **-all** argument is used.

```
-a
--all
```

Optional. Updates the enabled status for all the users. If this argument is present, no username should be specified.

```
-e <true|false>
--enabled=<true|false>
```

Optional. Specifies if the user should be enabled or disabled. The default value is **true**.

export-config

Exports a server configuration from the server database to the current working directory as a **configuration.xml** file.

SYNOPSIS

```
export-config [-f | --force] [-b value | --bootstrap-config=value] [-t
  value | --tool-password=value] [-h value | --hash=value] [export
  file]
```

OVERVIEW

Use this command to export a server configuration from the server database to a file. The configuration in the file can be edited and then imported back into the server database using the **import-config** command.

OPTIONS

```
-f
--force
```

Optional. Indicates that the tool should overwrite an existing destination file.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-h value
--hash=value
```

Optional. The (possibly abbreviated) hash of the configuration to export. Must consist of at least 6 hexadecimal characters.

```
[export file]
```

Optional. The path to the configuration file to create. The default value is **configuration.xml**.

export-ds-template

Exports the definition of a data source template.

SYNOPSIS

```
export-ds-template [-f | --force] [-c value | --configuration=value]
  [-b value | --bootstrap-config=value] <-n value | --name=value>
  [template definition file]
```

OVERVIEW

Use this command to export the definition of a data source template used by Information Services to a file.

OPTIONS

```
-f
--force
```

Optional. Indicates whether the tool should overwrite an existing destination file.

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-n value
--name=value
```

Required. The name of the data source template for which to export the definition.

```
[template definition file]
```

Optional. The path to the definition file to create. The default value is **template.xml**.

export-groups

Exports groups from the User Directory.

SYNOPSIS

```
export-groups [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-m <true|false> | --include-member-groups=
  <true|false>] [-u <true|false> | --include-member-users=<true|false>]
  [-g <true|false> | --include-guids=<true|false>] [-s <true|false> |
  --use-stdf=<true|false>] [-n <true|false> | --include-name-row=<true|
  false>] [export file] [-f | --force]
```

OVERVIEW

Use this command to export all groups from the User Directory. The exported groups can be imported on a different server.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-m <true|false>
--include-member-groups=<true|false>
```

Optional. Indicates whether the group hierarchy information (groups in groups) should be included. Can be used in conjunction with the **--include-member-users** argument to include all information. The default value is **false**.

```
-u <true|false>
--include-member-users=<true|false>
```

Optional. Indicates whether the group hierarchy information (users in groups) should be included. Can be used in conjunction with the **--include-member-groups** argument to include all information. The default value is **false**.

```
-g <true|false>
--include-guids=<true|false>
```

Optional. Indicates whether the Globally Unique Identifier (GUID) of each group should be included. The default value is **false**.

```
-s <true|false>
--use-stdf=<true|false>
```

Optional. Indicates whether the exported file should be created in Spotfire Text Data Format. If **false**, plain CSV format is used. The default value is **true**.

```
-n <true|false>
--include-name-row=<true|false>
```

Optional. Indicates whether the exported file should include a column name row. Applicable only when **--use-stdf** is set to **false**, because STDF always includes a name row. The default value is **false**.

```
[export file]
```

Optional. The path to the file to create. The default value is **groups.txt**.

```
-f
--force
```

Optional. Indicates that the tool should overwrite an existing destination file.

export-library-content

Exports content from the library.

SYNOPSIS

```
export-library-content [-f | --force] [-b value | --bootstrap-config=
  value] [-t value | --tool-password=value] <-p value | --file-path=
  value> <-u value | --user=value> [-a <true|false> |
  --include-access-rights=<true|false>] <-i value | --item-type=value>
  <-l value | --library-path=value>
```

OVERVIEW

Use this command to export content from the library.

OPTIONS

`-f`
`--force`

Optional. Indicates that the tool should overwrite any already existing file with the same name as specified in the path argument. All parts of the existing file (path.part0.zip, path.part1.zip, and so on) are also be deleted.

`-b value`
`--bootstrap-config=value`

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

`-t value`
`--tool-password=value`

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. Can be specified if and only if a password is given and **--enable-config-tool** argument is set to **true** (the default).

`-p value`
`--file-path=value`

Required. The file system path to where the item should be exported.

`-u value`
`--user=value`

Required. The user performing the export, should be a Library Administrator. The name of the user needs to include the user's domain name, for example **DOMAIN\user** or **user@domain**, unless the user is part of the configured default domain.

`-a <true|false>`
`--include-access-rights=<true|false>`

Optional. Specifies if access rights should be exported. The default value is **true**.

`-i value`
`--item-type=value`

Required. Which item types that should be exported from the library. It is possible to export all items, or all items of a certain type, from a folder. It is also possible to export a single item of a certain type. When exporting the content of a folder, valid values are: all_items, colorschemes, information_model, analysis_files and datafunctions. When exporting a single item, valid values are: analyticitem, dxpscript, bookmark, embeddedresource, query, join, dxp, datafunction, folder, colorscheme, column, datasource, filter and procedure.

```
-l value
--library-path=value
```

Required. The path in the library where the content is exported from. When exporting folder content, a path to the folder must be specified. When exporting a single item, a path to that specific item must be specified. The path must start with a slash (/). If the entire library should be exported the path should be **"/"**.

export-users

Exports users from the User Directory.

SYNOPSIS

```
export-users [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-i value | --include-password-hashes=value]
  [-s value | --use-stdf=value] [-g value | --include-guids=value] [-n
  value | --include-name-row=value] [export file] [-f | --force]
```

OVERVIEW

Use this command to export all users from the User Directory. The exported users can be imported on a different server.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-i value
--include-password-hashes=value
```

Optional. Indicates whether the exported file should include the password hashes. Passwords are relevant only if you use the Spotfire database for authentication. The default value is **false**.

```
-s value
--use-stdf=value
```

Optional. Indicates whether the exported file should be created in Spotfire Text Data Format. If **false**, plain CSV format is used. The default value is **true**.

```
-g value
--include-guids=value
```

Optional. Indicates whether the Globally Unique Identifier (GUID) of each user should be included. The default value is **false**.

```
-n value
--include-name-row=value
```

Optional. Indicates whether the exported file should include a column name row. Applicable only when **--use-stdf** is set to **false**, because STDF always includes a name row. The default value is **false**.

```
[export file]
```

Optional. The path to the file to create. The default value is **users.txt**.

```
-f
--force
```

Optional. Indicates that the tool should overwrite an existing destination file.

help

Displays the help overview or a specific help topic.

SYNOPSIS

```
help [topic name]
```

OVERVIEW

Use this command to display the help overview or a specific help topic.

OPTIONS

```
[topic name]
```

Optional. The name of the help topic to be displayed.

import-config

Imports a server configuration from a file to the server database.

SYNOPSIS

```
import-config [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-c value | --comment=value> [-d <true|false>
  | --delete-file=<true|false>] [import file]
```

OVERVIEW

Use this command to import a server configuration from a file to the server database and setting it as the current configuration. Such a server configuration file can be generated either by running the **export-config** command or by creating a new default configuration using the **create-default-config** command. If an identical configuration file already exist in the server database the existing configuration has its description updated.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-c value
--comment=value
```

Required. A comment describing the reason for the configuration change. Make sure to enclose the specified comment in quotes and to quote all special characters that might otherwise be consumed by the command line shell.

```
-d <true|false>
--delete-file=<true|false>
```

Optional. Indicates whether the imported configuration file should be deleted from the file system after a successful import. The default value is **'false'**.

```
[import file]
```

Optional. The path to the configuration file to import. The default value is **configuration.xml**.

import-groups

Imports groups to the User Directory.

SYNOPSIS

```
import-groups [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-m <true|false> | --include-member-groups=
  <true|false>] [-u <true|false> | --include-member-users=<true|false>]
  [-g <true|false> | --include-guids=<true|false>] [-n <true|false> |
  --has-name-row=<true|false>] [import file]
```

OVERVIEW

Use this command to import all groups in a given file to the User Directory. The groups can be imported including membership information or as a simple list.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-m <true|false>
--include-member-groups=<true|false>
```

Optional. Indicates whether the group hierarchy information (groups in groups) should be included. Can be used in conjunction with the **--include-member-users** argument to include all information. The default value is **false**.

```
-u <true|false>
--include-member-users=<true|false>
```

Optional. Indicates whether the group hierarchy information (users in groups) should be included. Can be used in conjunction with the **--include-member-groups** argument to include all information. The default value is **false**.

```
-g <true|false>
--include-guids=<true|false>
```

Optional. Indicates whether Globally Unique Identifiers (GUIDs) in the file should be included. The default value is **false**.

```
-n <true|false>
--has-name-row=<true|false>
```

Optional. Indicates whether the file contains a name row. Applicable only when the file is in plain CSV format because the Spotfire Text Data Format (STDF) always has a name row. The default value is **false**.

```
[import file]
```

Optional. The path to the file to import. The default value is **groups.txt**.

import-jaas-config

Imports new JAAS application configurations into the server configuration.

SYNOPSIS

```
import-jaas-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-f | --force] <-j value |
  --jaas-config-file=value>  [-n value | --name=value]
```

OVERVIEW

Use this command to import new JAAS application configurations into the server configurations.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-f
--force
```

Optional. Indicates that the JAAS application configurations should be imported into the server even if other configurations with the same names already exist. When this argument is enabled, the old configurations are overwritten.

```
-j value
--jaas-config-file=value
```

Required. The path to the JAAS application configuration file. The file is expected to be in the standard JAAS application configuration format

```
-n value
--name=value
```

Optional. The names of the JAAS application configurations to be imported into the server. Multiple names must be comma-separated and enclosed between quotes. If this argument is omitted, then all JAAS application configurations within the specified file is imported.

import-library-content

Imports content into the library.

SYNOPSIS

```
import-library-content [-b value | --bootstrap-config=value] [-t value
  | --tool-password=value] <-p value | --file-path=value> <-m value |
  --conflict-resolution-mode=value> <-u value | --user=value> [-e
  <true|false> | --prune-empty-directories=<true|false>] [-a <true|
  false> | --include-access-rights=<true|false>] [-i value |
  --item-type=value] [-l value | --library-path=value]
```

OVERVIEW

Use this command to import content into the library.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. Can be specified if and only if a password is given and **--enable-config-tool** argument is set to **true** (the default).

```
-p value
--file-path=value
```

Required. The file system path to the file that should be imported into the library. This should be the result of a previous library export and with a name ending with **.part0.zip**. If the export consists of several parts (ending with **.part1.zip** etc.) these must be placed in the same folder.

```
-m value
--conflict-resolution-mode=value
```

Required. Sets the conflict resolution mode that should be used if there is a conflict with already existing content in the library path given. The conflict resolution mode is applied for each conflicting item that is imported. Valid values are **KEEP_NEW**, **KEEP_OLD**, and **KEEP_BOTH**.

If you want to use the value **KEEP_BOTH** to copy a folder and the files it contains into a folder with the same name—for example, to copy the folder 3-7-14 from the folder structure ProjectA/Test/Deploy/3-17-14 to the Deploy folder in the folder structure ProjectA/Production/Deploy—you must first delete the Deploy folder from the second folder structure (in this case, the Production path). This is because Spotfire uses the combination of directory path and item type to help uniquely identify library items.

```
-u value
--user=value
```

Required. The user performing the import, should be a Library Administrator. Unless the user is part of the configured default domain, the name of the user needs to include the user's domain name, like **DOMAIN\user** or **user@domain**.

```
-e <true|false>
--prune-empty-directories=<true|false>
```

Optional. Specifies if empty directories should be created. The default value is **false**.

```
-a <true|false>
--include-access-rights=<true|false>
```

Optional. Specifies if access rights should be imported. The default value is **true**.

```
-i value
--item-type=value
```

Optional. Which item types that should be imported into the library. Valid values are: **all_items**, **colorschemes**, **information_model**, **analysis_files** and **datafunctions**. The default value is **all_items**.

```
-l value
--library-path=value
```

Optional. The path in the library where the content is imported. The path must specify an existing folder in the library. The default value is **/**.

import-users

Imports users to the User Directory.

SYNOPSIS

```
import-users [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-i <true|false> | --include-passwords=<true|
  false>] [-h <true|false> | --hash-passwords=<true|false>] [-g <true|
  false> | --include-guids=<true|false>] [-n <true|false> |
  --has-name-row=<true|false>] [import file]
```

OVERVIEW

Use this command to import all users in a given file to the User Directory. The users can be imported with or without passwords.

OPTIONS

`-b value`
`--bootstrap-config=value`

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

`-t value`
`--tool-password=value`

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

`-i <true|false>`
`--include-passwords=<true|false>`

Optional. Indicates whether passwords in the file should be included. The default value is **false**.

`-h <true|false>`
`--hash-passwords=<true|false>`

Optional. Indicates whether the included passwords should be hashed during import. Should be **false** if the users have previously been exported from a Spotfire Server because those passwords are already hashed. The default value is **false**.

`-g <true|false>`
`--include-guids=<true|false>`

Optional. Indicates whether Globally Unique Identifiers (GUIDs) in the file should be included. The default value is **false**.

`-n <true|false>`
`--has-name-row=<true|false>`

Optional. Indicates whether the file contains a name row. Applicable only when the file is in plain CSV format because the Spotfire Text Data Format (STDF) always has a name row. The default value is **false**.

`[import file]`

Optional. The path to the file to import. The default value is **users.txt**.

list-admins

Lists the server administrators.

SYNOPSIS

```
list-admins [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value]
```

OVERVIEW

Use this command to list the server administrators. Only direct members of the Administrator group are shown.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

list-auth-config

Displays the current authentication configuration.

SYNOPSIS

```
list-auth-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to display the current authentication configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

list-auth-mode

Displays the currently configured authentication mode.

This command is deprecated from 5.0 and is replaced by **list-auth-config**. For more information, see the “list-auth-config” on page 255 command.

SYNOPSIS

```
list-auth-mode [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to display the configured authentication mode.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

list-configs

Lists all available server configurations.

SYNOPSIS

```
list-configs [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-i | --include-incompatible] [-h value |
  --hash-abbrev=value]
```

OVERVIEW

Use this command to list the available configurations. The current configuration is indicated by an asterisk in the leftmost column.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-i
--include-incompatible
```

Optional. Indicates whether to include configurations incompatible with the current server version.

```
-h value
--hash-abbrev=value
```

Optional. The number of hexadecimal digits (between 6 and 40) to abbreviate the configuration hash to. The default value is **7**.

list-deployment-areas

Lists the deployment areas

SYNOPSIS

```
list-deployment-areas [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value]
```

OVERVIEW

Use this command to list the deployment areas as well as display what area that is the default deployment area.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

list-ds-template

Lists the data source templates.

SYNOPSIS

```
list-ds-template [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to list the data source templates.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

list-groups

Lists all groups.

SYNOPSIS

```
list-groups [-l value | --limit=value] [-s value | --search-expression=
  value] [-m | --list-members] [-b value | --bootstrap-config=value]
  [-t value | --tool-password=value]
```

OVERVIEW

Use this command to list all groups in the user directory.

OPTIONS

```
-l value
--limit=value
```

Optional. The maximum number of groups to list. The default value is **20**.

```
-s value
--search-expression=value
```

Optional. A search expression that can be used to search only for groups with names matching the expression.

```
-m value
--list-members
```

Optional. Determines whether or not to list the members.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

list-jaas-config

Lists the JAAS application configurations.

SYNOPSIS

```
list-jaas-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [--xml] [JAAS application configuration
  name]
```

OVERVIEW

Use this command to display the server JAAS application configurations. (It cannot display system JAAS application configurations.)

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
--xml
```

Optional. Specifies if the JAAS application configurations should be displayed in XML format, as it is stored within the **configuration.xml** file.

```
[JAAS application configuration name]
```

Optional. The names of the JAAS application configuration to display. Multiple names must be comma-separated and enclosed between quotes. If this argument is omitted, then all JAAS application configurations are displayed.

list-jmx-users

Lists all JMX users.

SYNOPSIS

```
list-jmx-users [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value]
```

OVERVIEW

Use this command to list all users who can access the server through JMX. The result contains the user name and the access level of each user.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

list-ldap-config

Displays LDAP configurations.

SYNOPSIS

```
list-ldap-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [--xml=value] [LDAP configuration id]
```

OVERVIEW

Use this command to display all LDAP configurations.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
--xml=value
```

Optional. Specifies that the LDAP configuration should be displayed in XML format instead of the standard JAAS application configuration format.

```
[LDAP configuration id]
```

Optional. Specifies the identifier of the LDAP configuration to be displayed. If no identifier is specified, then all LDAP configurations are displayed.

list-ldap-userdir-config

Lists the configuration for the User Directory LDAP mode.

SYNOPSIS

```
list-ldap-userdir-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to list the configuration for the User Directory LDAP mode.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

list-ntlm-auth

Displays the NTLM authentication service configuration.

SYNOPSIS

```
list-ntlm-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-S value | --server=value]
```

OVERVIEW

Use this command to display the NTLM authentication service configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-S value
--server=value
```

Optional. The name of the cluster server whose configuration should be displayed. If no name is specified, the global parameters common to all servers in the cluster are displayed.

list-online-servers

Lists all online servers.

SYNOPSIS

```
list-online-servers [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value]
```

OVERVIEW

Use this command to list all servers in the cluster that are currently online.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

OUTPUT

A table of all servers in the cluster that are currently online. An asterisk in the leftmost column is used to indicate that the server is the current *primus* server - responsible for handling tasks like synchronization of LDAP groups.

Example:

```
P  Server Name          IP Address    Version
   server1.example.com  192.0.2.1     6.5.0.70
*  server2.example.com  192.0.2.2     6.5.0.60
   server3.example.com  192.0.2.3     6.5.0.70
```

list-post-auth-filter

Displays the current Post Authentication Filter configuration.

SYNOPSIS

```
list-post-auth-filter [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to display the Post Authentication Filter configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

list-userdir-config

Lists the current User Directory configuration.

SYNOPSIS

```
list-userdir-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to list the current User Directory configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

list-userdir-mode

Lists the currently configured User Directory mode.

This command is deprecated and is replaced by **list-userdir-config**. See the “list-userdir-config” on page 263 command.

SYNOPSIS

```
list-userdir-mode [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to list the currently configured User Directory mode.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

list-users

Lists all users.

SYNOPSIS

```
list-users [-f | --force-synchronization] [-l value | --limit=value]
  [-s value | --search-expression=value] [-e <true|false> |
  --exclude-disabled=<true|false>] [-b value | --bootstrap-config=
  value] [-t value | --tool-password=value]
```

OVERVIEW

Use this command to list all users in the user directory. It does not work when using the User Directory Windows provider.

OPTIONS

```
-f
--force-synchronization
```

Optional. Indicates that the command should force a User Directory synchronization before attempting to list the users. This argument has no effect if the User Directory is running in database mode.

```
-l value
--limit=value
```

Optional. The maximum number of users to list. The default value is **100**.

```
-s value
--search-expression=value
```

Optional. A search expression that can be used to search only for users with names matching the expression.

```
-e value
--exclude-disabled=<true|false>
```

Optional. Indicates whether disabled users should be excluded. The default value is **false**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176

list-windows-userdir-config

Lists the configuration for the User Directory Window NT mode.

SYNOPSIS

```
list-windows-userdir-config [-c value | --configuration=value] [-b
  value | --bootstrap-config=value]
```

OVERVIEW

Use this command to list the configuration for the User Directory Windows NT mode.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

manage-deployment-areas

Manages the deployment areas

SYNOPSIS

```
manage-deployment-areas [-b value | --bootstrap-config=value] [-t value
  | --tool-password=value] [-R | --reset-all-group-areas] [-r |
  --reset-group-area] [-s | --set-group-area] [-c | --create-area] [-D
  | --delete-area] [-d | --default-area] [-g value | --group-name=
  value] [-a value | --area-name=value]
```

OVERVIEW

Use this command to change deployment area for groups, change default area, create and remove areas.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-R
--reset-all-group-areas
```

Optional. Use if all specified areas for all groups should be removed.

This does not affect the default area or any content on the areas. Users are using the default area after running this command. The **--reset-all-group-areas**, **--reset-group-area**, **--set-group-area**, **--create-area**, **--delete-area** and **--default-area** arguments are mutually exclusive.

```
-r
--reset-group-area
```

Optional. Use if an area for a specific group should be removed.

This does not affect the default area or any content on the area. If a user is not a member of any group with a specified area, the default area is used. The **--reset-all-group-areas**, **--reset-group-area**, **--set-group-area**, **--create-area**, **--delete-area** and **--default-area** arguments are mutually exclusive.

```
-s
--set-group-area
```

Optional. Use if an area should be set for a specific group. A user that is a member of this group gets access to the specified area instead of the default area. The **--reset-all-group-areas**, **--reset-group-area**, **--set-group-area**, **--create-area**, **--delete-area** and **--default-area** arguments are mutually exclusive.

```
-c
--create-area
```

Optional. Specifies that a new area should be created. The **--reset-all-group-areas**, **--reset-group-area**, **--set-group-area**, **--create-area**, **--delete-area** and **--default-area** arguments are mutually exclusive.

```
-D
--delete-area
```

Optional. Specifies that an existing area should be deleted. The **--reset-all-group-areas**, **--reset-group-area**, **--set-group-area**, **--create-area**, **--delete-area** and **--default-area** arguments are mutually exclusive.

```
-d
--default-area
```

Optional. Specifies that a the default area should be changed. The **--reset-all-group-areas**, **--reset-group-area**, **--set-group-area**, **--create-area**, **--delete-area** and **--default-area** arguments are mutually exclusive.

```
-g value
--group-name=value
```

Optional. The name of the group. Applicable for **--reset-all-group-areas**, **--reset-group-area**, and **--set-group-area**.

```
-a value
--area-name=value
```

Optional. The name of the area. Applicable for **--set-group-area**, **--create-area**, **--delete-area**, and **--default-area**.

modify-db-config

Modifies the common database connection configuration.

SYNOPSIS

```
modify-db-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-l value | --login-timeout=value] [-o
  value | --connection-timeout=value] [-i value | --min-connections=
  value] [-a value | --max-connections=value] [-p value |
  --pooling-scheme=value] [-q value] {-Ckey=value} [-e <true|false> |
  --clear-connection-properties=<true|false>]
```

OVERVIEW

Use this command to modify the common configuration for the connection to the Spotfire Server database. This configuration (which effects all servers) is merged with the configuration in the **bootstrap.xml** file on each server.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-l value
--login-timeout=value
```

Optional. The maximum time (in seconds) to wait for a connection to become available.

```
-o value
--connection-timeout=value
```

Optional. The maximum time (in seconds) a connection can stay idle in the connection pool before being closed and discarded.

```
-i value
--min-connections=value
```

Optional. The minimum number of connections to keep in the connection pool.

```
-a value
--max-connections=value
```

Optional. The maximum number of connections to keep in the connection pool.

```
-p value
--pooling-scheme=value
```

Optional. The connection pooling algorithm to be used. Valid values are:

- **WAIT**: The `--max-connections` parameter is strictly respected.
- **DYNAMIC**: The number of connections can occasionally exceed the configured maximum number.

```
-q value
```

Optional. An SQL query that should be run directly after a connection has been created.

```
-Ckey=value
```

Optional; can be specified multiple times with different keys. A JDBC connection property that is added to the existing list of connection properties. Several properties can be specified.

```
-e <true|false>
--clear-connection-properties=<true|false>
```

Optional. Clears the existing list of connection properties. The default value is **false**.

EXAMPLES

Setting the maximum number of connections in the pool:

modify-db-config --max-connections=100

Setting the pooling scheme:

modify-db-config --pooling-scheme=WAIT

Setting the size of the statement pool of the DataDirect driver:

modify-db-config -CMaxPooledStatements=20

modify-ds-template

Modifies a data source template.

SYNOPSIS

```
modify-ds-template [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-n value | --name=value> [-e <true|false>
  | --enable=<true|false>] [-r value | --rename=value] [-d value |
  --definition=value]
```

OVERVIEW

Use this command to modify a data source template used by Information Services.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-n value
--name=value
```

Required. The name of the data source template to modify.

```
-e <true|false>
--enable=<true|false>
```

Optional. Indicates whether the data source template should be enabled. If no argument is given, the value is unchanged.

```
-r value
--rename=value
```

Optional. The name to rename the data source template to. If no argument is given the value is unchanged.

```
-d value
--definition=value
```

Optional. The path to the file to containing a new data source template definition. If no argument is given the value is unchanged.

promote-admin

Assigns full administrator privileges to a user.

SYNOPSIS

```
promote-admin [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-u value | --username=value>
```

OVERVIEW

Use this command to promote a user to administrator by making the user account a member of the Administrator group.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-u value
--username=value
```

Required. The name of the user to be promoted to administrator. Unless the user is part of the configured default domain, the name of the user needs to include the user's domain name, like **DOMAIN\user** or **user@domain**.

remove-ds-template

Removes a data source template.

SYNOPSIS

```
remove-ds-template [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-n value | --name=value>
```

OVERVIEW

Use this command to remove a data source template.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-n value
--name=value
```

Required. The name of the data source template to remove.

remove-jaas-config

Removes the specified JAAS application configurations from the server configuration.

SYNOPSIS

```
remove-jaas-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-n value | --name=value>
```

OVERVIEW

Use this command to remove JAAS application configurations from the server.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-n value
--name=value
```

Required. The names of the JAAS application configurations to be removed from the server. Multiple names must be comma-separated and enclosed between quotes.

remove-ldap-config

Removes LDAP configurations.

SYNOPSIS

```
remove-ldap-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <LDAP configuration ids>
```

OVERVIEW

Use this command to remove LDAP configurations.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
<LDAP configuration ids>
```

Required. Specifies a comma-separated list of identifiers of the LDAP configurations to be removed.

remove-license

remove a license from a group.

SYNOPSIS

```
remove-license <-g value | --group=value> <-l value | --license=value>
  [-b value | --bootstrap-config=value] [-t value | --tool-password=
  value]
```

OVERVIEW

Use this command to remove a license from a group.

OPTIONS

```
-g value
--group=value
```

Required. The group to have its licenses removed.

```
-l value
--license=value
```

Required. The license to remove.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

run

Runs a configuration script.

SYNOPSIS

```
run <script file>
```

OVERVIEW

Use this command to run a configuration script.

OPTIONS

```
<script file>
```

Required. The name of the script to be executed.

SCRIPT SYNTAX

Each line must contain the name of a command and its arguments. Arguments can be quoted using either single or double quotes. Lines beginning with a hash character (**#**) are regarded as comments and have no effect. Lines ending with a backslash character (****) are continued on the next line with the backslash character removed before parsing.

The special script command "`echo`" can be used to echo messages to the console. See “Script Language” on page 52.

s3-download

Downloads the data of library items in Amazon S3 storage.

SYNOPSIS

```
s3-download [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-t value | --tool-password=value] <-i
  value | --items=value> <-d value | --destination=value>
```

OVERVIEW

Used this command to download the data of library items in Amazon S3 storage.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-i value
--items=value
```

Required. A comma-separated list of the library items (GUIDs) to download.

```
-d value
--destination=value
```

Required. The directory where the downloaded items should be saved.

set-auth-mode

Sets the authentication mode.

This command is deprecated from 5.0 and is replaced by **config-auth**. See “config-auth” on page 202 command.

SYNOPSIS

```
set-auth-mode [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-a value | --auth-method=value> [-d |
  --jaas-database] [-l | --jaas-ldap] [-w | --jaas-windows] [-j value |
  --jaas-custom=value]
```

OVERVIEW

Use this command to set the authentication mode.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-a value
--auth-method=value
```

Required. The authentication method to use. The following methods are supported: BASIC, CLIENT_CERT, NTLM, Kerberos and External. The names can be specified in either upper or lower case.

```
-d
--jaas-database
```

Optional. Use the Spotfire database authentication source, as configured in the SpotfireDBLogin JAAS application configuration. This option is permitted only with the BASIC authentication method. It is also mutually exclusive with all other options related to BASIC authentication sources.

```
-l
--jaas-ldap
```

Optional. Use the LDAP authentication source, as configured in the SpotfireLDAP JAAS application configuration. This option is permitted only with the BASIC authentication method. It is also mutually exclusive with all other options related to BASIC authentication sources.

```
-w
--jaas-windows
```

Optional. Use the Windows NT authentication source, as configured in the SpotfireWindows JAAS application configuration. This option is permitted only with the BASIC authentication method. It is also mutually exclusive with all other options related to BASIC authentication sources.

```
-j value
--jaas-custom=value
```

Optional. Use the custom JAAS application configuration with the specified name. This option is permitted only with the BASIC authentication method. It is also mutually exclusive with all other options related to BASIC authentication sources.

set-config

Sets the current server configuration.

SYNOPSIS

```
set-config [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-h value | --hash=value> <-c value |
  --comment=value>
```

OVERVIEW

Use this command to set the current configuration to one of the existing configurations. Refer to “list-configs” on page 256.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-h value
--hash=value
```

Required. The (possibly abbreviated) hash of the configuration to set. Must be at least the first six hexadecimal characters of the hash.

```
-c value
--comment=value
```

Required. A comment describing the reason for the configuration change.

set-db-config

Sets the common database connection configuration.

SYNOPSIS

```
set-db-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] [-l value | --login-timeout=value] [-o
  value | --connection-timeout=value] [-i value | --min-connections=
  value] [-a value | --max-connections=value] [-p value |
  --pooling-scheme=value] [-q value] {-Ckey=value}
```

OVERVIEW

Use this command to set the common configuration for the connection to the Spotfire Server database. This configuration (which effects all servers) is merged with the configuration in the **bootstrap.xml** file on each server.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-l value
--login-timeout=value
```

Optional. The maximum time (in seconds) to wait for a connection to become available. The default value is **10**.

```
-o value
--connection-timeout=value
```

Optional. The maximum time (in seconds) a connection can stay idle in the connection pool before being closed and discarded. The default value is **600**.

```
-i value
--min-connections=value
```

Optional. The minimum number of connections to keep in the connection pool. The default value is **5**.

```
-a value
--max-connections=value
```

Optional. The maximum number of connections to keep in the connection pool. The default value is **40**.

```
-p value
--pooling-scheme=value
```

Optional. The connection pooling algorithm to be used. Valid values are:

- **WAIT**: The `--max-connections` parameter is strictly respected.
- **DYNAMIC**: The number of connections can occasionally exceed the configured maximum number.

The default value is **WAIT**.

```
-q value
```

Optional. An SQL query that should be run directly after a connection has been created.

```
-Ckey=value
```

Optional; can be specified multiple times with different keys.

A JDBC connection property. Several properties can be specified.

EXAMPLES

Example of how to set the maximum number of connections in the pool:

```
set-db-config  --max-connections=100
```

Example of how to set the pooling scheme:

set-db-config --pooling-scheme=WAIT

Example of how to set the size of the statement pool of the DataDirect driver:

set-db-config CMaxPooledStatements=20

set-license

Sets a license and license functions for a group.

SYNOPSIS

```
set-license <-g value | --group=value> <-l value | --license=value> [-f
  value | --functions=value] [-b value | --bootstrap-config=value] [-t
  value | --tool-password=value]
```

OVERVIEW

Use this command to set a license and license functions for a group.

OPTIONS

```
-g value
--group=value
```

Required. The group that should get the licenses set.

```
-l value
--license=value
```

Required. The license to set.

```
-f value
--functions=value
```

Optional. The license functions to enable.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

set-restart-policy

Sets the server restart policy.

SYNOPSIS

```
set-restart-policy [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-p value | --policy=value>
```

OVERVIEW

Use this command to set the way the server(s) react to configuration changes. Each server periodically checks for configuration changes and handles any such changes according to the policy set using this command.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-p value
--policy=value
```

Required. Controls whether the server(s) should be restarted when a configuration change is detected, and if so when. Valid values are:

- **MANUAL**: The changes do not have an effect until the server(s) are manually restarted
- **AUTOMATIC_FORCE**: The server(s) are immediately automatically restarted
- **AUTOMATIC_ON_IDLE**: The server(s) are automatically restarted when considered idle

that the **AUTOMATIC_FORCE** option can result in currently running user operations being aborted.

set-userdir-mode

Sets the User Directory mode.

This command is deprecated from 5.0 and is replaced by **config-userdir**. See “config-userdir” on page 226 command.

SYNOPSIS

```
set-userdir-mode [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <-m value | --mode=value>
```

OVERVIEW

Use this command to set the User Directory mode.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-m value
--mode=value
```

Required. The name of the User Directory mode to use. Supported values are **database**, **ldap** and **windows**.

show-basic-ldap-auth

Shows the LDAP authentication source for use with the BASIC authentication method.

SYNOPSIS

```
show-basic-ldap-auth [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to show the LDAP authentication source(s) for use with the BASIC authentication method. The configuration is stored within the SpotfireLDAP JAAS application configuration.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

show-config-history

Shows the configuration history.

SYNOPSIS

```
show-config-history [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-h value | --hash-abbrev=value]
```

OVERVIEW

Use this command to show the configuration history. The most recent entry is the current configuration.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-h value
--hash-abbrev=value
```

Optional. The number of hexadecimal digits to abbreviate the configuration hash to. Must be a number between 6 and 40. The default value is **7**.

show-deployment

Shows the current deployment.

SYNOPSIS

```
show-deployment [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-a value | --area=value] [-s | --show-ids]
```

OVERVIEW

Use this command to show the current deployment in a given area.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-a value
--area=value
```

Optional. The deployment area for which to show the current deployment. If no area is specified, the deployment of the default area is showed.

```
-s
--show-ids
```

Optional. Indicates whether the package IDs should be included in the output. A package ID is needed to remove a specific package using the update-deployment command. For more information, see “update-deployment” on page 287.

show-import-export-directory

Shows the library import/export directory.

SYNOPSIS

```
show-import-export-directory [-c value | --configuration=value] [-b
  value | --bootstrap-config=value]
```

OVERVIEW

Use this command to display the library import/export directory. All library import and export operations are done from and to this directory, which can be a local directory or can reside on a shared disk.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

show-join-database

Shows the configured default join database.

SYNOPSIS

```
show-join-database [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

Use this command to show the configured default join database, used by Information Services.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

show-library-permissions

Shows permissions set in the library.=.

SYNOPSIS

```
show-library-permissions [-b value | --bootstrap-config=value] [-t
  value | --tool-password=value] <-l value | --library-path=value> [-r
  <true|false> | --recursive=<true|false>] [-x <true|false> |
  --expand-groups=<true|false>] [-d <true|false> | --downward=<true|
  false>] [-p value | --path-to-report=value] [-f <true|false> |
  --force-overwrite=<true|false>]
```

OVERVIEW

This command creates a report file that shows the permissions in the library.

Permissions are set on directories, if no permission is set the directory inherits the permissions from the directory above.

You can use this command in three different ways:

- It can show if any permissions are set explicitly on a directory.
- It can show what permissions are effective on a certain directory. If no permissons are set on the directory, it continues to read upwards until it finds the directory from which the permissions are inherited (see **recursive** option).
- It can be used to report on all directories with permissions explicitly set in a branch of the directory (see the **downward** option).

The resulting file should be possible to read in TIBCO Spotfire. It has headers that explain the display in the different columns.

This command might take some time to run. Also you might need to increase the Java memory allocation to run the command, especially if the users are displayed.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. Can be specified if a password is given and **--enable-config-tool** argument is set to **true** (the default).

```
-l value
--library-path=value
```

Required. The path in the library to start to report with (must start with a **/**).

```
-r <true|false>
--recursive=<true|false>
```

Optional. If no permission is set on this directory, continue upwards until permissions are found. The default value is **false**.

```
-x <true|false>
--expand-groups=<true|false>
```

Optional. Specifies whether groups are expanded to show their members. The default value is **false**.

Members of the Administrator and Library Administrator group can see all content. When expand-groups is **true**, these implicit rights are also taken into account, and these groups and their members are also displayed.

```
-d <true|false>
--downward=<true|false>
```

Optional. Lists permissions on an entire branch of the library, and shows only folders where permissions are set explicitly. (This option takes precedence over the recursive option.) The default value is **false**.

```
-p value
--path-to_report=value
```

Optional. The name of the report file that should be generated. If not provided, an automatic name is generated.

```
-f <true|false>
--force-overwrite=<true|false>
```

Optional. If a name for the report file is provided but a file with that name already exists, set this option to **'true'** to overwrite the existing file. The default value is **'false'**.

show-licenses

Shows licenses set on the server.

SYNOPSIS

```
Show-licenses [-b value | --bootstrap-config=value]
  [-t value | --tool-password=value] [-l value | --license=value]
  [-x <true|false> | --expand-groups=<true|false>]
  [-p value | --path-to_report=value
  [-f <true|false> | --force-overwrite=<true|false>]
```

OVERVIEW

Use this command to create a report file that shows the licenses set on the server.

You can read the resulting file in TIBCO Spotfire: The file has headers that explain the contents displayed in the columns. The column "From Group" contains the group on which the license is explicitly set. For every group that has a license set explicitly, the resulting groups and users (if the expand option is set) are shown once.

Users get the sum of all licenses (and functions). When you analyze the file, note that a user and a license might occur more than once if the user gets its licenses from more than one group with explicit licenses set.

This command can take some time to run. Also, you might need to increase the Java memory allocation to run the command, especially if the users are displayed.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to "bootstrap.xml" on page 176.

```
-l value
--license=value
```

Optional. An optional, comma-separated list of licenses. If provided, the report contains only these licenses. If an invalid entry is given, the valid licenses are displayed.

```
-x <true|false>
--expand-groups=<true|false>
```

Optional. Should groups be expanded to show their users. The default value is 'false'.

```
-p value
--path-to_report=value
```

Optional. The name of the report file that should be generated. If not provided, an automatic name is generated.

```
-f <true|false>
--force-overwrite=<true|false>
```

Optional. If a name for the report file is provided but a file with that name already exists, set this option to **'true'** to overwrite the existing file. The default value is **'false'**.

show-restart-policy

Shows the server restart policy.

SYNOPSIS

```
show-restart-policy [-c value | --configuration=value] [-b value |
  --bootstrap-config=value]
```

OVERVIEW

This command shows the restart policy. Valid values are as follows:

- **'MANUAL'** -- Changes have no effect until the server(s) are manually restarted.

- **'AUTOMATIC_FORCE'** -- The server(s) are immediately automatically restarted.
- **'AUTOMATIC_ON_IDLE'** -- The server(s) are automatically restarted when considered idle.
  The **'AUTOMATIC_FORCE'** option can result in currently-running user operations being aborted.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

switch-domain-name-style

Switches the domain names for all users and groups from one style (DNS or NetBIOS) to the other (for all configured domains).

SYNOPSIS

```
switch-domain-name-style [-b value | --bootstrap-config=value] [-t
  value | --tool-password=value] <-n value | --new-domain-name-style=
  value>
```

OVERVIEW

Use this command to switch the domain names for all existing users and groups from one style (DNS or NetBIOS) to the other (for all configured domains). The new domain name style must first be configured using the **config-userdir** command. that this command is only applicable when using a User Directory in LDAP mode against Active Directory.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-n value
--new-domain-name-style=value
```

Required. The new domain name style. Valid values are **dns** and **netbios**.

test-jaas-config

Tests a JAAS application configuration.

SYNOPSIS

```
test-jaas-config [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] [-c value | --configuration=value] <-j value |
  --jaas-configuration=value> <-u value | --username=value> [-p value |
  --password=value]
```

OVERVIEW

Use this command to test a JAAS application configuration by performing a log in attempt, using the specified credentials. It can test either a configuration stored in the server database or a configuration stored in an exported configuration file. To test a configuration stored in a configuration file, use the **--configuration** argument. Otherwise the configuration stored in the database is tested. If the JAAS log in module requires a connection to the server database, the **--configuration** argument cannot be used.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-c value
--configuration=value
```

Optional. The path to an exported server configuration file. If this parameter is omitted, the application attempts to retrieve the configuration parameters from the server database using the file **bootstrap.xml**, specified by the **--bootstrap** argument.

```
-j value
--jaas-configuration=value
```

Required. The name of the JAAS application configuration to test.

```
-u value
--username=value
```

Required. The name of the user to log in as.

```
-p value
--password=value
```

Optional. The password of the user to log in as. If the password is omitted, the command prompts the user for it on the console.

update-deployment

Updates the current deployment.

SYNOPSIS

```
update-deployment [-b value | --bootstrap-config=value] [-t value |
  --tool-password=value] <-a value | --area=value> [-c | --clear] [-r
  value | --remove-packages=value] [-v value | --version=value] [-d
  value | --description=value] [-f | --force-update] [deployment files]
```

OVERVIEW

Use this command to add a new deployment or to update the current deployment in a given area.

OPTIONS

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
-t value
--tool-password=value
```

Optional. The configuration tool password used to decrypt the database password in the file **bootstrap.xml**. If the tool password is omitted, the command prompts the user for it in the console. Refer to “bootstrap.xml” on page 176.

```
-a value
--area=value
```

Required. The deployment area that should be updated.

```
-c
--clear
```

Optional. Indicates that all existing packages should be removed before any new files are added. If no files to add to the deployment are provided, the deployment area is empty.

```
-r value
--remove-packages=value
```

A comma separated list of IDs of packages that should be removed from the deployment. The IDs can be determined using the 'show-deployment' command. Should not be specified together with the **--clear** argument.

```
-v value
--version=value
```

Optional. The version of the new deployment. If no value is given it is taken from the current deployment, or from the last added distribution if one is added.

```
-d value
--description=value
```

Optional. The description of the new deployment. If no value is given it is taken from the current deployment, or from the last added distribution if one is added.

```
-f
--force-update
```

Optional. Indicates that users connecting to the server should be forced to update their clients.

```
[deployment files]
```

Optional. A comma separated list of files (packages and distributions) that should be added to the deployment. that the paths cannot contain spaces.

update-ldap-config

Updates LDAP configurations.

SYNOPSIS

```
update-ldap-config [-c value | --configuration=value] [-b value |
  --bootstrap-config=value] <--id=value> [-t value | --type=value] [-s
  value | --servers=value] [--clear-context-names] [-n value |
  --context-names=value] [-u value | --username=value] [-p value |
  --password=value] [--schedules=value] [--clear-schedules]
  [--user-search-filter=value] [--user-name-attribute=value]
  [--authentication-attribute=value] [--security-authentication=value]
  [--referral-mode=value] [--request-control=value] [--page-size=value]
  [--import-limit=value] [--user-display-name-attribute=value]
  [--group-display-name-attribute=value] {-Ckey=value}
```

OVERVIEW

Use this command to update LDAP configurations.

OPTIONS

```
-c value
--configuration=value
```

Optional. The path to the server configuration file. The default value is **configuration.xml**.

```
-b value
--bootstrap-config=value
```

Optional. The path to the bootstrap configuration file. See the **bootstrap.xml** help topic for more information about this file.

```
--id=value
```

Required. Specifies the identifier for the LDAP configuration to be updated.

```
-t value
--type=value
```

The type of LDAP server. The following names are valid types:

- ActiveDirectory
- SunOne
- SunJavaSystem
- Custom

When you specify any of the first three types, a type-specific configuration template is automatically applied in runtime so that the most fundamental configuration options are configured automatically.

When you specify a **Custom** LDAP server type, there is no such configuration template and all those configuration options must be specified explicitly. When a custom LDAP configuration is to be used for authentication or with the User Directory LDAP provider, the **--user-search-filter** and **--user-name-attribute** arguments must be specified. For such an LDAP configuration to be used for group synchronization, additional parameters must also be specified when running the **config-ldap-group-sync** command.

```
-s value
--servers=value
```

Specifies a whitespace-separated list of LDAP server URLs. An LDAP server URL has the format **<protocol>://<server>[:<port>]**:

- **<protocol>**: Either **LDAP** or **LDAPS**
- **<server>**: The fully qualified DNS name of the LDAP server.
- **<port>**: Optional. Number indicating the port number the LDAP service is listening on. When using the LDAP protocol, the port number defaults to **389**. When using the LDAPS protocol, the port number defaults to **636**. Active Directory LDAP servers also provides a Global Catalog containing forest-wide information, instead of domain-wide information only. The Global Catalog LDAP service by default listens on port number **3268** (LDAP) or **3269** (LDAPS).

The Spotfire Server does not expect any search base, scope, filter or other additional parameters after the port number in the LDAP server URLs. Such properties are specified using other configuration options for this command.

Unless the **--discover** option is used, this argument is mandatory.

Examples: LDAP server URL

LDAP://myserver.example.com

LDAPS://myserver.example.com

LDAP://myserver.example.com:389

LDAPS://myserver.example.com:636

LDAP://myserver.example.com:3268

LDAPS://myserver.example.com:3269

```
--clear-context-names
```

Optional. Clears context names from the LDAP configuration. This argument can be used together with the **--context-names** argument to remove all old context names before adding the new.

```
-n value
--context-names=value
```

Optional. A list of distinguished names (DNs) of containers holding LDAP accounts to be visible within the Spotfire Server. When specifying more than one DN, then the DNs must be separated by pipe-characters (**|**). The specified context names are added to the context names that are already configured. To set the context names from scratch, use the **--clear-context-names** argument with the **--context-names**.

If the specified containers contain a large number of users, of which only a few should be visible in the Spotfire Server, a custom user search filter can be specified to include only the designated users (see the **--user-search-filter** argument).

Examples:

CN=users,DC=example,DC=com

OU=project-x,DC=research,DC=example,DC=com

```
-u value
--username=value
```

Optional. The name of the LDAP service account to be used when searching for users (and optionally also groups) in the LDAP server. This service account does not need to have any write permissions, but it needs to have read permissions for all configured context names (LDAP containers). For most LDAP servers, the account name is the account's distinguished name (DN). For Active Directory, the account name can also be specified in the forms **ntdomain\name** and **name@dnsdomain**.

Examples:

CN=spotsvc,OU=services,DC=research,DC=example,dc=COM

RESEARCH\spotsvc (Active Directory only)

spotsvc@research.example.com (Active Directory only)

```
--password=value
```

Optional. The password for the LDAP service account.

```
--schedules=value
```

Optional. A comma-separated list of schedules for when the LDAP synchronization should be performed. The schedules are given in a cron-compatible format, where each schedule consists of either five fields or one shorthand label. Make sure to enclose the value in double quotes. The specified schedules are added to the schedules that are already configured. To set the schedules from scratch, use the **--clear-schedules** argument with the **--schedules**.

The five fields are, from left to right, with their valid ranges: minute (0-59), hour (0-23), day of month (1-31), month (1-12) and day of week (0-7, where both 0 and 7 indicate Sunday). A field can also be configured with the wildcard character *, indicating that any moment in time matches this field. A group synchronization is triggered when all fields match the current time. If both day of month and day of week have non-wildcard values, then only one of them has to match.

There are also the following shorthand labels that can be used instead of the full cron expressions:

@yearly or @annually: run once a year (equivalent to 0 0 1 1 *)

@monthly: run once a month (equivalent to 0 0 1 * *)

@weekly: run once a week (equivalent to 0 0 * * 0)

@daily or @midnight: run once a day (equivalent to 0 0 * * *)

@hourly: run once an hour (equivalent to 0 * * * *)

@minutely: run once a minute (equivalent to * * * * *)

@reboot or @restart: run every time the Spotfire Server is started

Refer to the Wikipedia overview article on the cron scheduler.

`--clear-schedules`

Optional. Clears from the LDAP configuration the LDAP synchronization schedules. This argument can be used together with the **--schedules** argument to remove all old schedules before adding the new.

`--user-search-filter=value`

Optional; must be specified for custom LDAP configurations, either when running this command or the **create-ldap-config** command.

Specifies an LDAP search expression filter to be used when searching for users.

- The parameter is mandatory for all custom configurations.
- For Active Directory servers, the parameter value defaults to **objectClass=user**.
- For any version of the Sun Directory Servers, it defaults to **objectClass=person**.

If only a subset of all the users in the specified LDAP containers should be allowed access to the Spotfire Server, a more detailed user search filter can be used. The search expression can for instance be expanded so that it also puts restrictions on which groups the users belong to, or which roles they have.

- For Active Directory servers, access can be restricted to only those users belonging to a certain group by using a search expression with the pattern **&(objectClass=user)(memberOf=<groupDN>)**, where **<groupDN>** is to be replaced by the real DN of the group to which the users must belong. If the users are divided among multiple groups, use the pattern **&(objectClass=user)(|(memberOf=<firstDN>)(memberOf=<secondDN>))**. Add extra **(memberOf=<groupDN>)** sub-expressions as needed.

  Active Directory Example:
  &(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)

- For a Sun Java System Directory Server version 6 and later, the same effect as above can be achieved by using a search expression with the pattern **&(objectClass=person)(isMemberOf=<groupDN>)**. If the users are divided among multiple groups, use the pattern **&(objectClass=person)(|(isMemberOf=<firstDN>)(isMemberOf=<secondDN>))**. Add extra **(isMemberOf=<groupDN>)** sub-expressions as needed.

  Sun Java System Directory Server Example:
  &(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)

- For Sun ONE Directory Servers as well as the newer Sun Java System Directory Servers or the older iPlanet Directory Server, access can be restricted to only those users having certain specific roles. The search expression for role filtering must match the pattern **&(objectClass=person)(nsRole=<roleDN>)**. If multiple roles are of interest, use the pattern **&(objectClass=person)(|(nsRole=<firstDN>)(nsRole=<secondDN>)**. Add extra **(nsRole=<roleDN>)** sub-expressions as needed.

  Sun ONE Directory Servers Example: **&(objectClass=person)(isMemberOf=cn=project-x,dc=example,dc=com)**

The syntax of LDAP search expression filters is specified by the RFC 4515 document. Consult this documentation for information about more advanced filters.

`--user-name-attribute=value`

Optional; must be specified for custom LDAP configurations, either when running this command or the **create-ldap-config** command.

Specifies the name of the LDAP attribute containing the user account names. For Active Directory servers the value defaults to **sAMAccountName**. For a Sun Java System Directory Server (or any older Sun ONE Directory Server or iPlanet Directory Server) with a default configuration, it defaults to **uid**.

`--authentication-attribute=value`

Optional; should be used only for advanced setups. It is not set by default.

Specifies the name of the LDAP attribute containing a user identity that can be used for binding (authenticating) to the LDAP server. This attribute fills no purpose in most common LDAP configurations, but can be useful in more advanced setups, where the distinguished name (DN) does not work for authentication or where users should be able to log in using a username that does not map directly to an actual LDAP account.

When setting up SASL with DIGEST-MD5 in an Active Directory environment, the DN does not work for authentication and the **userPrincipalName** attribute must be used instead. The **--authentication-attribute** argument should then be set to **userPrincipalName** and the **--user-name-attribute** argument should be set to **sAMAccountName** (the latter value also happens to be the default value for an Active Directory LDAP configuration, so there's no need to set it explicitly). See also the **--security-authentication** argument.

When setting up SASL with GSSAPI in an Active Directory environment, the DN does not work for authentication and the **sAMAccountName** or **userPrincipalName** attribute must be used instead. The **--authentication-attribute** argument should then be set to **sAMAccountName** or **userPrincipalName** and the **--user-name-attribute** argument should be set to **sAMAccountName** (the latter value also happens to be the default value for an Active Directory LDAP configuration, so there is no need to set it explicitly). See also the **--security-authentication** argument.

Example:

- By setting the **--user-name-attribute** argument to **cn** and the **--authentication-attribute** argument to **userPrincipalName** in an Active Directory environment, the users can log in to the Spotfire Server using their CN attribute values, but underneath the hood, the Spotfire Server actually uses the **userPrincipalName** attribute value of the LDAP account with the matching CN for the actual authentication.

`--security-authentication=value`

Optional; should be used only in advanced setups. The default value is **simple**.

This parameter specifies the security level to use when binding to the LDAP server.

- To enable anonymous binding, it should be set to **none**.
- To enable plain username/password authentication, it should be set to **simple**.
- To enable SASL authentication, it should be set to the name of the SASL mechanism to be used, for instance **DIGEST-MD5** or **GSSAPI**. Use multiple **-C** arguments to set the additional JNDI environment properties that the SASL authentication mechanism typically requires.

When setting up SASL with DIGEST-MD5 in an Active Directory environment, all accounts must use reversible encryption for their passwords. This is typically not the default setting for the domain controller. The **--authentication-attribute** argument must also be used to specify the **userPrincipalName** attribute for the actual authentication to work correctly.

When setting up SASL with GSSAPI in an Active Directory environment, the **--authentication-attribute** argument must be used to specify either the **sAMAccountName** or the **userPrincipalName** attribute and the custom property **kerberos.login.context.name** must be mapped to the JAAS application configuration **SpotfireGSSAPI**. This in turn requires a fully working Kerberos configuration file at **<installation directory>/jdk/jre/lib/security/krb5.conf**.

`--referral-mode=value`

Optional. Specifies how LDAP referrals should be handled. Valid arguments are **follow** (automatically follow any referrals), **ignore** (ignore referrals) and **throw** (fail with an error). The default and recommended value is **follow**.

`--request-control=value`

Optional. Determines the type of LDAP controls to be used when executing search queries to the LDAP server. The default behavior is to probe the LDAP server for the best supported request control. The paged results control is always preferred, because it provides the most efficient way of retrieving the query result set. The virtual list view control can also be used for the same purpose if the paged results control is not supported. The virtual list view control is automatically used together with a sort control. Both the paged results control and the virtual list view control supports a configurable page size, set by the **--page-size** argument.

- To explicitly configure the server for probing, set the argument value to **probe**.
- To configure the server for the paged results control, set the argument value to **PagedResultsControl**.
- To request the virtual list view control, set the argument value to **VirtualListViewControl**.
- To completely disable request controls by setting the argument value to **none**.

The default value is **probe**.

`--page-size=value`

Optional. Specifies the page size to be used with the paged results control or the virtual list view control when performing search queries to the LDAP server. The page size value defaults to **2000** for both the paged results control and the virtual list view control.

`--import-limit=value`

Optional. Specifies a threshold that limits the number of users that can be imported from an LDAP server to the Spotfire Server in one query. This can be used to prevent accidental flooding of the Spotfire Server User Directory when integrating with an LDAP server with tens or even hundreds of thousands of users. By setting an import limit, the administrator can be sure that an unexpected high number of users do not affect the server performance. By default, there is no import limit. To explicitly request unlimited import, set the parameter value to **-1**. All positive numbers are treated as an import limit. In most cases, it is recommended to leave this parameter untouched.

`--user-display-name-attribute=value`

Optional. Specifies the name of the LDAP attribute containing the user display names.

`--group-display-name-attribute=value`

Optional. Specifies the name of the LDAP attribute containing the group display names.

`-Ckey=value`

Optional; can be specified multiple times with different keys.

Specifies additional JNDI environment properties to be used when connecting to the LDAP server. that it does not add to the previously configured custom properties, it replaces them completely. If you want to keep any of the old custom properties, make sure to specify them once again when adding new ones.

Example: The equivalent of specifying the **--security-authentication=DIGEST-MD5** argument is `-Cjava.naming.security.authentication=DIGEST-MD5`

Example: Updating the context names

update-ldap-config --id="ldap1" --context-names="OU=project-x,DC=research,DC=example,DC=com|OU=phbs,DC=management,DC=example,DC=com"

version

Displays the current version of the server.

SYNOPSIS

```
version
```

OVERVIEW

Use this command to display the current version of the server.

Reference: Troubleshooting

Spotfire Server Logs

Logs store the important diagnostic information about the Spotfire Server that can help in troubleshooting and resolving issues. Spotfire server, by default runs at the minimal logging level and this can be elevated, when needed.

The most **important** log is the "server.log" (previously named as dss.log in 3.x versions). This log file stores information about all activities on the server and can be very handy in troubleshooting issues.

If you encounter an issue with Spotfire Server, provide the server logs to Spotfire Support when you log the support request.

Here is a list of logs that are available with Spotfire Server, categorized based on what they capture:

Log Name	File Name	Contents
Configuration Tool Log	tools.log	This log file stores information about activity of the configuration tool / Configuration Command Line Tool. For example, if you run any configuration commands at the command prompt or use the UI, this is the log that captures that information.
Server Log	server.log	Information about all activity on the server except those events recorded in the Server Access Log.
Server Access Log	access.log	Information about client access and access attempts to the server and files in the library.
Server Usage Log	usage.log	Information about client access and access attempts to the server.
Library Log	library.log	Information about Spotfire Library usage.
Library Import/Export Log	impex.log	Information about Spotfire Library imports and exports.
Information Services Usage Log	isusage.log	Information about Information Services usage.

SQL Log	sql.log	Information about executed SQL queries performed when an information link is executed.
SOAP Log	soap.log	Information about SOAP communication.
Server Diagnostics Log	server-diagnostics.log	Diagnostic information about server measures.
Startup Log	startup.log	Information about JAR files loaded on server startup.

Different logging levels that can be used for Server logs:

Spotfire Server runs by default at **Info (log4j.properties)** logging level. This logging level can be elevated to capture more information about issues, errors etc. There are two methods of changing the logging level:

1 **When Spotfire Server is up and running** - using the *Open Logs and Diagnostics* tab from Spotfire Server > Welcome Page.

2 **When Spotfire Server is not running** - by modifying the **com.spotfire.logging.config.file** parameter in **web.xml** file located under **<Spotfire Server Install Dir.>\tomcat\webapps\spotfire\WEB-INF\** folder

Elevated logging is useful for troubleshooting; however, you should run Spotfire Server in **Info (log4j.properties)** logging mode when server is running fine.

You can use many logging methods in Spotfire; however, the following are the most commonly used methods for troubleshooting Spotfire server issues.

- log4j.properties:
  The default log level set in Spotfire Server. Captures the events in **Info** mode.
- log4j-debug.properties:
  When this log level is set, the Server Log (**server.log**) logs detailed debug information, as well as warnings, errors, and other information. The SQL Log (**sql.log**) logs detailed SQL information. If the server is started from a command prompt or shell, the output to the command prompt or shell is also included in the Server Log.
- log4j-trace.properties:
  This level gives more detailed information than the DEBUG level and should be used only when needed. that this logging level is very verbose and running Spotfire Server.

Location of server logs:

- Location of Spotfire Server Log files:
  - Spotfire Server logs are located under **<Spotfire Server Install Dir.>\tomcat\logs** folder.

Example: C:\tibco\tss\6.5.0\tomcat\logs

- Location of Upgrade Log files:
  - Spotfire Server Upgrade logs are located under **<Spotfire Server Install Dir.>\tools\upgrade\logs** folder

    Example: C:\tibco\tss\6.5.0\tools\upgrade\logs
- Logs default directory location can be changed by modifying the following parameter in the **<Spotfire Server Install Dir.>\tomcat\webapps\spotfire\WEB-INF\web.xml** file

```
<context-param>
  <param-name>log.dir</param-name>
        <param-value>/../../logs</param-value>
  </context-param>
```

Enabling debug logging on Spotfire Server:

When Spotfire Server is up and running:

1 Launch Spotfire Administration Console by using Spotfire Server URL in a Web browser

2 Click **Open Logs and Diagnostics**.

3 Log in using Administrator credentials and click on **Server Log Files** tab.

4 Select the **required log** from the dropdown menu to the left and set the **required logging level** using the dropdown menu to the right on the screen.

5 Click **Refresh** to see the latest entries in the log.

Enabling debug logging from this console does not require a server restart.

To export the log file from this console, click the **Export Log File** icon in the right upper corner.

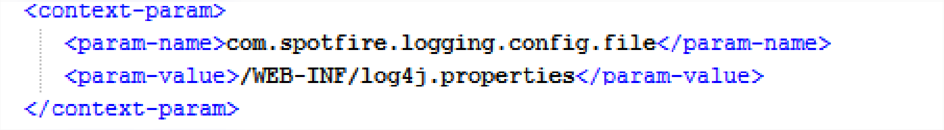
Based on requirements, select a different logging level and a different log file using the various options in the dropdown menu options

Information under the *Diagnostics* tab provides useful diagnostic information about various aspects of Spotfire Server like **Application Server**, **Database Server**, **Uptime** etc.

When Spotfire Server is not working:

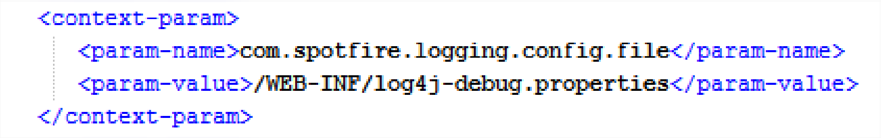
1 **Back up** and open the **web.xml** file from **<Spotfire Server Install Dir.>\tomcat\webapps\spotfire\WEB-INF** folder in a text editor (for example pad).

2. Find **log4j.properties parameter** in this file. An example on how this parameter looks like the **web.xml** file:



```
<context-param>
    <param-name>com.spotfire.logging.config.file</param-name>
    <param-value>/WEB-INF/log4j.properties</param-value>
</context-param>
```

3. Replace it with **log4j-debug.properties** and save the file. Here is how the changed parameter should look like:



```
<context-param>
    <param-name>com.spotfire.logging.config.file</param-name>
    <param-value>/WEB-INF/log4j-debug.properties</param-value>
</context-param>
```

4. Save the file.

5. Restart the "Spotfire Server Service" from "Windows Services" for the changes to take effect.

Always take a backup of the web.xml file before making any modifications

Use any text editor (for example pad) to modify the XML files. Do not use applications such as Wordpad, which can change the file encoding and result in corrupted XML Fles.

Disable **Debug** logging after the troubleshooting is completed. We do not recommend running the server in debug mode for longer periods.

It is a good practice to back up the existing logs and clear the logs folder before capturing the debug logs.

Capturing Spotfire Server Diagnostics:

1. Launch **Spotfire Administration Console** by using Spotfire Server URL in a Web browser.

2. Click on **Open Logs and Diagnostics**.

3. Login using Administrator credentials and click the *Diagnostics* tab.

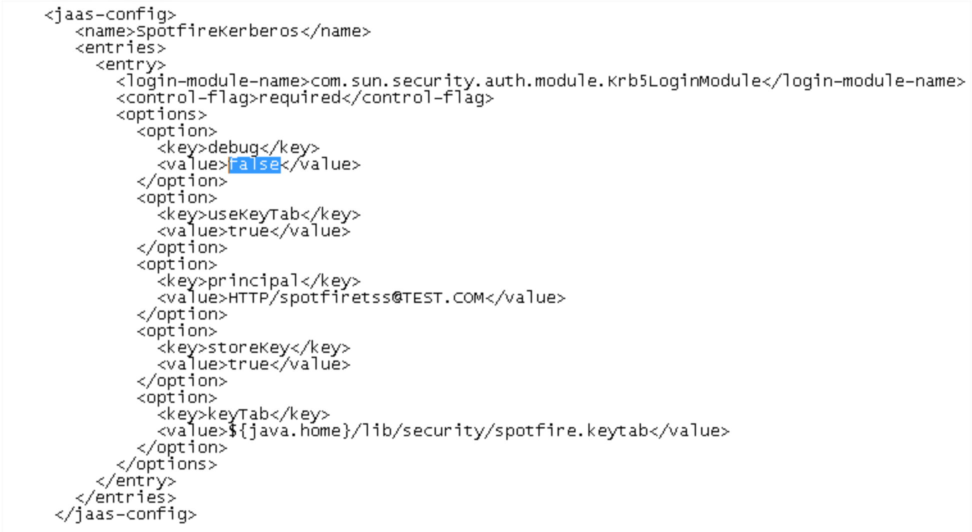
4. Click **Export to file** and save the file.

Enabling Kerberos Debug logging on Spotfire Server:

If there are any issues with the Kerberos authentication, please follow the instructions below to enable Kerberos debug logging on Spotfire Server:

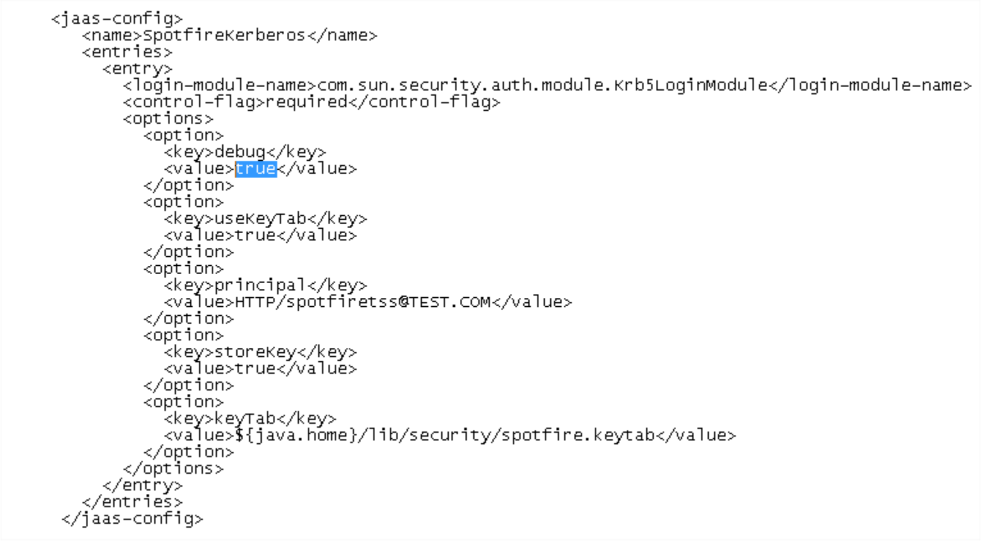
1. Open the **configuration.xml** file from **<Spotfire Server Install Dir.>\tomcat\bin** folder in a text editor (ex: pad)

2 Locate the following configuration block in the **configuration.xml** file:



```
<jaas-config>
   <name>SpotfireKerberos</name>
   <entries>
     <entry>
       <login-module-name>com.sun.security.auth.module.Krb5LoginModule</login-module-name>
       <control-flag>required</control-flag>
       <options>
         <option>
           <key>debug</key>
           <value>false</value>
         </option>
         <option>
           <key>useKeyTab</key>
           <value>true</value>
         </option>
         <option>
           <key>principal</key>
           <value>HTTP/spotfiretss@TEST.COM</value>
         </option>
         <option>
           <key>storeKey</key>
           <value>true</value>
         </option>
         <option>
           <key>keyTab</key>
           <value>${java.home}/lib/security/spotfire.keytab</value>
         </option>
       </options>
     </entry>
   </entries>
 </jaas-config>
```

3 Change the value for **debug** key from **false** to **true**.



```
<jaas-config>
   <name>SpotfireKerberos</name>
   <entries>
     <entry>
       <login-module-name>com.sun.security.auth.module.Krb5LoginModule</login-module-name>
       <control-flag>required</control-flag>
       <options>
         <option>
           <key>debug</key>
           <value>true</value>
         </option>
         <option>
           <key>useKeyTab</key>
           <value>true</value>
         </option>
         <option>
           <key>principal</key>
           <value>HTTP/spotfiretss@TEST.COM</value>
         </option>
         <option>
           <key>storeKey</key>
           <value>true</value>
         </option>
         <option>
           <key>keyTab</key>
           <value>${java.home}/lib/security/spotfire.keytab</value>
         </option>
       </options>
     </entry>
   </entries>
 </jaas-config>
```

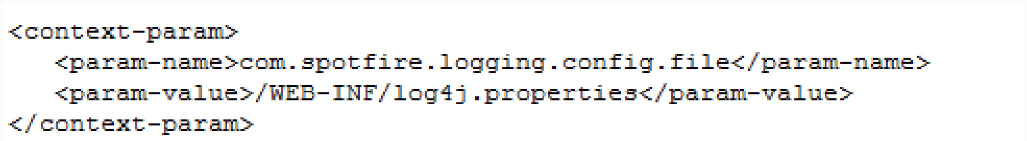
4 Save the file.

5 Launch a Command prompt on Spotfire Server and browse to the **<Spotfire Server install Dir.>\tomcat\bin** folder.

6 Import the configuration using **import-config** command. For example: **config import-config -comment="Enabled Kerberos Debug Logging"**

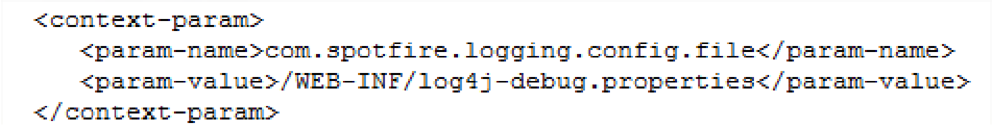
7 Open the **web.xml** file from **<Spotfire Server Install Dir.>\tomcat\webapps\spotfire\WEB-INF\** folder in a text editor (for example pad).

8. Find the **log4j.properties** parameter in this file. An example on how this parameter looks like the **web.xml** file:



```
<context-param>
   <param-name>com.spotfire.logging.config.file</param-name>
   <param-value>/WEB-INF/log4j.properties</param-value>
</context-param>
```

9. Replace it with **log4j-debug.properties** and save the file. Here is how the changed parameter should look like:



```
<context-param>
   <param-name>com.spotfire.logging.config.file</param-name>
   <param-value>/WEB-INF/log4j-debug.properties</param-value>
</context-param>
```

10. Save the file.

11. Restart the “Spotfire Server Service” from 'Windows Services” for the changes to take effect.

Always take a **backup** of the **web.xml** file before making any modifications.

Use any text editor (for example pad) to modify the XML files. Do not use applications such as Wordpad, which can change the file encoding and result in corrupted XML Files.

Disable **Debug** logging after the troubleshooting is completed. We do not recommend running the server in debug mode for longer periods.

It is a good practice to back up the existing logs and clear the logs folder before capturing the debug logs.

Basic Troubleshooting Steps

If there are any issues with Spotfire Server, here are a few aspects that needs to be checked:

1. **Spotfire Database**

   - Make sure Spotfire Server Database is up and running.
   - Validate the Database credentials specified in the **bootstrap.xml** file.
     - Ensure the DB user has access to all the required Spotfire database Tables and Procedures i.e. if you log in to Spotfire Server Database with those credentials, the user should be able to browse and access all the contents of Spotfire database.
   - Make sure there is communication between Spotfire Server machine and Spotfire database Server -Ex: Ping DB Server from Spotfire Server.
   - Review the Spotfire Server logs for more clues.

2. **Spotfire Server**

- Make sure that Spotfire Server has network connectivity.
- Spotfire Server Service is up and running.
  - If a Custom User Account is used to run the Spotfire Server Service, ensure the account credentials are valid and not locked.
- No port conflicts with the Spotfire Server ports.
- Spotfire Server Administration Console can be accessed outside the Spotfire Server machine.
  - If it works fine on the server machine but not accessible outside the server, make sure there is no Firewall or Proxy blocking the Server access.
- If "Spotfire Administration Console" comes up but fails to authenticate, check the server logs for more clues.

Memory Dumps

Occasionally there might be problems were memory is exhausted, this usually shows as an out-of-memory exception in the log, but can also manifest itself as a deadlock if you are using Microsoft SQL Server. The first step is to increase memory, see "Modifying the Virtual Memory" on page 154. If the problem still exists Spotfire Support might want to get a dump of the memory to see if there is any memory leak. When you are running the server as a Windows service, it is complicated to create a memory dump. For a simpler alternative, you can navigate to a page that creates a memory dump.

Memory dumps contain the entire state of the running server and can thus contain sensitive information.

When a memory dump is created, the Java Virtual Machine halts for a short period. Therefore, there are some extra steps required to enable this, it can only be done and read by someone who has access to the server's file system and also is a member of the Administrator group; it is not sufficient to be part of the Diagnostics Administrator group.

1. Navigate with your web browser to the **Server Start Page > Open Logs and Diagnostics > Troubleshooting > Create Memory Dump**.

2. You need to prove that you have access to the server itself by creating a "proof file" with a specific random name on the file system of the server. A new name is generated every time the server is restarted or when a memory dump has been made. The name of the "proof file" is shown on the page and it does not proceed until the file exists. The file does not have to have any content. The purpose is only to show that the user not only is Administrator but also has write access to the file system on the sever.

3. After the "proof file" is in place, the heap dump can be done by navigating back to the page, or by clicking the **Reload** link. A memory dump is created. This can take some time. Any previous dump file is overwritten. When it is completed, the path to the file on the server's file system is displayed. You must go to the server to retrieve the file; there is no download functionality on the page. After you have analyzed the file, delete

it: It can contain sensitive information. On normal termination of the server, the generated heap dump file is deleted automatically.

There is an advanced setting to disable the functionality altogether. This requires you to manually edit the **configuration.xml** and enter a new node in the configuration.xml (**tools > enable-memory-dump** with the value “false”) and then make sure that the configuration is uploaded and made active.

Thread Dumps

To help troubleshoot cases when the server either seems to be hanging or when things take an unusual amount of time, a dump of thread activity can help Spotfire Support to determine what is happening. When the server is running as a Windows service, it is somewhat complicated to create this thread dump. For a simpler alternative, navigate to the page that can create a thread dump.

From your web browser, navigate to the **Server Start Page > Open Logs and Diagnostics > Troubleshooting > View Thread Dump**.

The dump displays a short stack trace of all the running threads, along with information about whether they are waiting for something.

Troubleshooting Bundle

To ensure that support can start working on a support case, we have provided a utility that creates a zip archive, which you can then send to Spotfire Support. This zip archive contains the following:

- The entire logs directory
- A thread dump
- The results of diagnostics

To create such a bundle navigate with your web browser to the **Server Start Page > Open Logs and Diagnostics > Troubleshooting > Generate Troubleshooting Bundle**. This can take a while, but when done your browser should start to download the generated zip file.

Common Issues

1 **Issue**:

Symptom:

Spotfire Server can fail to start with the following error message: *“Error initializing the Spotfire web application. Please contact the server administrator”* and the following errors are captured in the server logs:

============

EVERE: Catalina.start:

LifecycleException: service.getName(): "Spotfire"; Protocol handler start failed: java.net.BindException: Address already in use: JVM_Bind <null>:

============

Resolution:

This is an indication of a Port conflict. You can check if any of the Spotfire Server ports are blocked by other processes on the Spotfire Server machine. Either stop those services so that Spotfire Server can grab these ports or assign a different port by modifying the **server.xml** file located under **\tomcat\conf** folder

2 **Issue**:

Symptom:

Spotfire Server can run out of JVM memory, which can cause issues like Spotfire Server failure/hang, unable to allow any new connections or throw the following message when opening up any files:

============

Error message: Java heap space

============

The following errors can be captured in the server logs:

============

Caused by: java.lang.OutOfMemoryError: GC overhead limit exceeded

……..

SEVERE: Exception invoking periodic operation:

java.lang.OutOfMemoryError: Java heap space

============

Resolution:

This exception is thrown by the garbage collector in the underlying Java and is not specific to Spotfire. This error essentially means that you need to add more memory. Refer to the section “Modifying the Virtual Memory” on page 154 for more information.

3 **Issue**:

Symptom:

User's can not be able to log in to Spotfire Professional or WebPlayer clients. Administrators can fail to log into Spotfire Administration Console.

Server logs can indicate the following LDAP error code:

=======

javax.naming.AuthenticationException: [LDAP: error code 49 - 80090308: LdapErr: DSID-0C090334, comment: AcceptSecurityContext error, data 52e, vece]

=========

Resolution:

The LDAP error code indicates that the log incredentials used for LDAP binding are invalid. One of the main reasons this can happen is if the password of the LDAP Service Account is expired. To resolve this issue, modify the LDAP configuration with the updated credentials.

4 **Issue**:

Symptom:

Users can not be able to log into Spotfire Professional or WebPlayer clients. Administrators can fail to log into Spotfire Administration Console.

Server logs can indicate the following LDAP error code:

```
=======

javax.naming.AuthenticationException: [LDAP: error code 49 - 80090308: LdapErr: DSID-0C0903A9, comment: AcceptSecurityContext error, data 533, v1db1 ]

=========
```

Resolution:

The LDAP error code indicates that the Service Account that is used for LDAP binding can be locked out/disabled. To resolve this issue, enable the Service Account and then try again.